Facial Feature Processing Using Artificial Neural Networks

Nicholas David Porter

A dissertation submitted in satisfaction of the requirements for the degree of Doctor of Philosophy

University of Warwick

Department of Engineering

February 1998

Contents

Abbreviations and Symbols

Mathematical conventions

The following conventions have been adopted for the presentation of mathematical information within this thesis:

x	The variable x.
$\mathbf{x}$	The vector $\mathbf{x}$.
x_i	The ith element of vector $\mathbf{x}$.
$\mathbf{x}^\top$	The transpose of the vector $\mathbf{x}$.
$\mathbf{X}$	The matrix $\mathbf{X}$.
$\mathbf{x}_j$	The j row of the matrix $\mathbf{X}$.
x_{ji}	The element at the jth row and ith column of matrix $\mathbf{X}$.

In some instances the row and column indices for matrices are reversed. This is made clear by the definitions in the relevant sections:

Abbreviations

The following is a list of the abbreviations and symbols used in this thesis along with their meaning.

a	Activation: the result of the weighted sum of inputs to a neuron.
ANN	Artificial Neural Network.
α	Learning rate.
bpp	Bits per pixel.

CMY	The cyan, magenta, yellow colour encoding scheme.
CMYK	The cyan, magenta, yellow, black colour encoding scheme.
E	Network error.
E_i	Error from node i.
E_p	Network error with pattern p on the inputs.
$f()$	A network transfer function, usually sigmoidal.
GA	Genetic Algorithm
HSV	The hue, saturation, value colour encoding scheme.
MLP	Multilayer Perceptron.
r	The learning signal in the general learning rule (see 3.4.1).
RBFN	Radial Basis Function Network.
RGB	The red, green, blue colour encoding scheme.
SOM	Self-Organising Map.
σ	The standard deviation of a population or function.
$\text{sgn}(x)$	The sign function of x. Returns a 1 if x is positive or -1 if x is negative.
t_i	The desired (target) output of node i.
$\mathbf{t}$	The desired (target) output vector.
τ	Time.
TLU	Threshold Logic Unit.
$\mathbf{w}_i^\tau$	The weight vector on the inputs to node i at time τ.
w_{ij}^τ	The weight on the connection from node or input j to node i at time τ.
x_{ij}	Input j to node i.
$\mathbf{x}$	The input vector to a layer.
y_i	The output from node i.
$\mathbf{y}$	The output vector from a network.

Glossary

Below is a list of some of the terminology used in this thesis and their meaning:

activation function	The activation function is the mathematical relationship between the weighted sum of the inputs of a neuron and its output.
classification	The process of classification determines to which of a fixed number of groups or *classes* a given data element belongs.
cluster	A series of data elements which have some property in common which groups them together to the exclusion of other members of the data set. Often clusters will be based on some distance measure between the data elements.
cluster membership function	A function which defines which members of a population of vectors belong to which of a number of clusters spanning the data space.
error surface	The error in the output of a neural network may be viewed as a function of the weights in the network. If the network has only two weights then this can be represented as a surface, i.e. the error plotted against the values of the two weights. This is extended to situations with n weights such that the error may be thought of as a n-dimensional surface when plotted against the weights.
full face	A full face image is one where the person is looking straight at the camera. This is the most common form of image used in facial image processing

generalisation — The property of a classifier where it is able to correctly classify data that was not part of the original data set used in the training process or for setting up the rules governing classification. This is normally a desirable property of a classifier.

hard limiter function — A hard limiter function is one that takes a real value as its input and produces a bi-valued output, often 0 or 1, depending on the relationship between the input value and some threshold.

hyperplane — An n-dimensional hyperplane is a plane in n-dimensional space. In neural network classifiers, often the n-dimensional pattern space is divided up by hyperplanes to produce the classification regions.

normalise — The process of normalisation has many different variations. It refers to re-scaling data in some manner such that it conforms to a particular standard. For example the data may be re-scaled such that its possible values lie within a given range, or an image may be re-scaled to eliminate variations in size. In this thesis, each occurrence of a normalisation technique is explained within its context.

pattern — In neural network terms, a pattern is a single input vector used for either training or testing the network. This term is often interchangeable with the word *vector*.

profile — A profile image of a face is one taken with the camera looking at the side of the head, i.e. the person is facing a direction perpendicular to the camera's line of sight. Another standard position is $\frac{3}{4}$ profile where the person is looking towards the camera at an angle of 45° to the camera's line of sight.

target — Target can have on of two meanings in this thesis. In the context of facial recognition, the target is the correct identification of the face to be recognised. In the neural network context, the target is the desired output of the network associated with any given input pattern.

testing data — A subset of the total data set that is used to evaluate the performance of a neural network or other classifier. A good classification accuracy resulting from the use of the testing data set shows that the classifier has *generalised* from the training data that was presented to it. The performance of the network with testing data is often used as an indication of when the training should be stopped.

training data — A subset of the total data set that is used in the training of a neural network or other classifier.

validation data — A subset of the total data set that is independant of the training and testing data sets. It is used to confirm the results from the neural network in a similar manner to the testing data. The difference is that the vallidation data plays no part in the training process and therefore gives the best indication of how well the network has generalised.

List of Figures

List of Tables

List of Algorithms

Acknowledgements

The author would like to thank the following for their contributions to this work:

The Police Service Research and Development Group at the UK Home Office for the proposal of the work undertaken in preparing this thesis, the provision of funds for the thesis and the supply of data for carrying out the experimental work.

The Engineering and Physical Sciences Research Council for their funding of the work.

Dr. E. Hines for his supervision of the work in this thesis.

Prof. D. J. Whitehouse for his helpful comments on the structure and content of this thesis and suggestions on the direction of the work.

Dr. M. Craven, Dr. A. H. Khan, Dr. A. B. Larkin and Dr. A. C. Pardoe for friendship and support during the course of the study for this thesis.

Declaration

The work reported on in this thesis was produced by the author unless otherwise stated. All work reported here was carried out during the author's period of study while registered as a PhD student at the University of Warwick. A publication resulting from work leading to this thesis is presented in Appendix C.

Abstract

Describing a human face is a natural ability used in eveyday life. To the police, a witness description of a suspect is key evidence in the identification of the suspect. However, the process of examining "mug shots" to find a match to the description is tedious and often unfruitful. If a description could be stored with each photograph and used as a searchable index, this would provide a much more effective means of using "mug shots" for identification purposes.

A set of *descriptive measures* have been defined by Shepherd[73] which seek to describe faces in a manner that may be used for just this purpose. This work investigates methods of automatically determining these descriptive measures from digitised images.

Analysis is performed on the images to establish the potential for distinguishing between different categories in these descriptions. This reveals that while some of the classifications are relatively linear, others are very non-linear.

Artificial neural networks (ANNs), being often used as non-linear classifiers, are considered as a means of automatically performing the classification of the images. As a comparison, simple linear classifiers are also applied to the same problems.

The work shows that there is potential in the use of ANNs for the classification of facial features. On simple features such as the presence or absense of a beard, high accuracies between 90 and 100% were acheived. The more complex classification of eye colour yielded lower results of around 30% accuracy and further analysis is performed to show the limitations in the data sets that have produced this result.

Overall, the work shows that there is potential for ANNs techniques in this area though much further development would be needed before all of the classifications are at a standard that could be used by the police.

Chapter 1

Introduction

When a witness to a crime is interviewed by the police, one of the objectives of the interview is to identify the suspected criminal. One part of this is to make use of information held, either in paper form or on computer, regarding people with previous criminal convictions. The traditional system employed in UK Police stations for witness identification of suspects from this information could be considered rather primitive and has a low success rate[79]. Police photos of convicted criminals are kept in albums and identifying a suspect from these is simply a matter of looking through the album until a match is found. This is an ineffective system as after seeing many photographs[43], the witness is no-longer able to make correct judgement in identifying the suspect and may be misled by similar looking photographs.

What is needed is a method of reducing the number of photographs that have to be presented to the witness and thus increasing the probability that the suspect will be presented to the witness while they are still able to correctly identify them from within the set of other photographs. One method is to group the photographs by crime type. This will help in the case of crimes that have few offenders but will not aid with "common" crimes where the number of offenders in a given region is still in the 1000's.

Another approach is to group faces according to likeness and only show the witness photographs which are similar to their description of the suspect. The difficulty that arises in performing this selection is how to measure similarity i.e. which features should be used to perform the grouping. In addition, it is likely that the photographs

would need to be re-grouped for each witness presentation. That is, it is likely that one "similar" grouping that was used for a given witness would not be of use for a different suspect identification. If this sorting and grouping were performed manually then the work involved would be prohibitive so some form of automated system is required.

To this end, a computerised system has been developed for presenting photographs to witnesses based on their descriptions[73]. This is described in detail in Section 2.2. The system is based on a series of 50 descriptive measures that are stored in association with each facial image. These are used as an index to the library of photographs so that searches can be made for faces that match a given description. The workload here is now governed by the efforts required in the generation of the descriptive measures. This has been performed for some small databases[74] of upto 2000 faces using manual techniques[72] but such manual data entry is not suitable for adding descriptions to hundreds of thousands of records that make up the whole of the database stored on the UK Police National Computer. Rather what is needed is an automated method of generating the descriptive measures.

Given that the images can be scanned into a digital format for computer entry, the task becomes one of interpreting these digital images to yield the descriptive measures. This is the focus of work contained in this thesis; it may be summarised in the statement:

> "Given a digitised colour image of a face, extract from that image a series of pre-defined descriptive measures."

This work is part of an ongoing project by the UK Home Office Police Science Research and Development Group which has awarded a number of research contracts to various groups as part of the overall project. The work in this thesis is specifically concerned with the potential application of artificial neural networks to the problem of classifying facial features. The remainder of this introduction chapter will give an overview of the problem and methods used in attempting to solve it.

1.1 What's in a Face?

Given that most faces are made up from the same basic components - two eyes, two ears, one nose, one mouth and some hair on top - with the same basic arrangement, there is a remarkable degree of variation that allows us to easily recognise familiar individuals in the midst of a crowd. Various questions arise such as "what are the variables that allow humans to perform this recognition?" "Are there measurable quantities that can be found that will allow an automatic discrimination between faces?" Psychologists have been examining these and other similar questions for many years.

There is an interesting phenomena observed in the recognition problem, in that Europeans find it easier to distinguish between other Europeans than between those of other races and vice versa[75]. This demonstrates that regular exposure to one "kind" of face gives the human observer an ability to more easily distinguish within that group. However studies have shown[22, 23] that there is nothing inherent in the Asian face, for example, that makes Asians any more like each other than Europeans, i.e. the degree of variation in the features is the same.

One can think of many instances where some form of automated recognition system would be beneficial such as, for example, building security systems[71]. In designing such systems, there can be great benefit in considering the psychological processes involved in human recognition. For example, the phenomena described above is one that is observed with the artificial neural networks described in the next section. That is, repeated presentation of a given subset of the available data when training the network gives a greater degree of discrimination within that subset than within larger data set. This type of phenomena is an example of how artificial neural network systems are mimicking the behaviour of their biological counterparts. However the systems that are in existence today are still poor in their abilities when compared to human recognition.

In examining the various automated systems that have been developed for processing face images, a number of key processes can be identified as common to them all. Possibly the most important aspect of facial image processing is that of either generating a "standard" image or developing methods to overcome variations. For example the

human recognition ability is not hampered by differing facial expressions but these can make large differences to measurements taken from points on the face; the difference between a smile and a frown can greatly alter distances measured from the corners of the mouth. For this reason, automated recognition experiments will often be performed using "expressionless" faces. The data used in this thesis is also of this type with the faces contrained in position within the image and showing no expression. Chapter 4 details the data used in this thesis.

Other factors that produce variation in facial images are the lighting conditions, the background and the orientation of the head with respect to the camera. In many recognition experiments, these will therefore be controlled by, for example, the use of camera lights, plain coloured backgrounds and fixed positions of the subject's head with respect to the camera. However, research has also been directed specifically at these issues, in particular Govindaraju *et al.*[26] have worked on identifying faces in cluttered images while Beymer *et al.*[4], Lando *et al.*[42] and others have looked at position and rotation invariance in facial recognition.

While various researchers have looked at individual problems in facial image processing, each solution has been specific to a particular problem rather than being generic. There would appear to be much complex information stored in the image of a face which current processing techniques have yet to fully make use of or extract.

Chapter 2 will discuss in much greater detail the work done by other researchers in facial image processing and show the extent of current research.

1.2 Intelligent Computers?

Artificial intelligence (AI) has developed into a fairly sizable branch of computing with the aim of using computing techniques to reproduce the intelligence observed in biological systems. One area of AI involves the study of biological neural systems and attempts to mimic their behaviour by the use of various technologies such as electronics, computing and optics. These systems are commonly known as Artificial Neural Networks.

Artificial Neural Networks (ANNs) have been applied successfully to many pat-

tern recognition problems and are an area of increasing research activity. Although sometimes shrouded in mystery, the field of ANNs is one that is now of interest to many different communities looking for an alternative method of solving complex data processing problems.

The inspiration behind ANNs is the biological neural system, that is, man is seeking after an artificial implementation of the abilities that come naturally to biological systems. Man has observed the ways in which biological systems develop and learn as they grow older and wishes now to use such properties in artificial systems. That is we wish to have electronic systems that are able to learn for themselves in the same manner that, for example, a baby learns to walk and talk. The baby has no knowledge of how the process of walking should be performed he or she simply works it out by trial and error with guidance from various teachers until it becomes a natural action. Similar situations occur in various data processing problems where there is no knowledge of a mathematical or analytical solution to the problem. An alternative method of finding a solution is needed. The existence of "learning" systems brings about the possibility of solving these problems which defy analytical methods due to their complexity or perceived limitations in terms of, for example, the type of information required.

In addition to learning, biological systems have a natural ability to generalise, for example a child does not have to be shown every kind of dog in order to know what a dog is. Just a few examples are sufficient and from that the child generalises when meeting other dogs. This feature of the biological neural system can be one of great benefit in some data processing problems. For example, in classification applications, it means that only a small percentage of the data population need be used in the "training" process where the neural system learns about the task it is to perform. This brings great potential to ANNs based systems in that many problems exist where only a small percentage of the total data set is available for use in training.

In seeking to emulate the power of biological neural systems, it is important to consider what it is that the neural network is doing and thus one may eliminate certain situations where the ANNs technique is of no benefit compared with others and indeed may perform worse. In considering the biological source for these systems, it may

be seen that different neural networks do not necessarily give responses that agree with each other. For example human descriptions of a given colour can vary wildly from one observer to another. From considering this kind of property of biological neural systems, it may be expected that similar behaviour will be experienced with the artificial counterparts. That is independently trained ANNs are likely to give different results when presented with the same input data.

For this reason a neural network system is not suitable to be applied to a problem where accurately predictable results are needed i.e. the system is required to always give the same response rather than being dependent on the particular training cycle. This form of problem falls more into the area of algebra where a "proved" result can be given. However, there are instances where it is not possible to produce algebraic expressions or algorithms for the classification that is required. It is in instances such as these that ANNs can be particularly powerful.

1.3 The Classification Problem

In data processing, it is common to want to classify a given piece of data as belonging to one of a number of classes. For example in a traffic census, vehicles may be classified according to type (car, lorry, bus, van etc.). In this particular case the classification is usually performed by a human observer. However, there are many cases where it would be desirable that the classification process is automated in some manner. For example, in the traffic census, suppose the human observer were replaced by a video camera. Then the input data is the image from the camera and a classification is to be made based on that image. This could be automated given sufficient *a priori* knowledge about the appearance of each class of vehicle and suitable image processing algorithms.

This form of problem may be generalised by stating that in all classification problems, the input data will consist of a series of vectors each describing one object to be classified. Certain rules must then be followed to determine the classification. If the input vectors are single valued, then the classification may be as simple as applying a set of thresholds, i.e., the total range of input values is split into a number of groups; for example, values between 0 and 3 belong to class $\mathcal{A}$, values between 3 and 4 to class

$\mathcal{B}$ and values above 4 to class $\mathcal{C}$. However, usually the input vectors are multi-valued and some more complex method will be used to assign the correct classes.

One method frequently used is that of Euclidean distance measures. A sample of data known as the "training data" for which the correct classification is known is used to establish the centres of the each class, that is, taking all training data vectors belonging to a given class, the mean value is found. Then for each of the vectors that requires classification, the Euclidean distance between that vector and each of the "cluster centres" is found according to

$$d = \sqrt{(\mathbf{x} - \mathbf{c})^2} \; . \tag{1.1}$$

where $\mathbf{x}$ is the vector in question, $\mathbf{c}$ is the centre of the cluster and d is the Euclidean distance. The vector is then said to belong to the cluster whose centre it is closest to i.e. the one where the Euclidean distance is the smallest.

This is a simple method of classification which presumes that data classes are arranged within the data space in groups that can be separated from each other. Unfortunately, this is often not the case and in these situations, more complex classification methods are required. For example, distance measures can be modified to fit the shape of the data classes that do not follow spherical groupings around a central point; this is explained further in Section 5.1.1.

A classification problem may be thought of as a mapping function where a given input vector is to be mapped to a certain output vector. When classification is considered in this light artificial neural networks may be considered as a method of solving such problems that are too complex for simple Euclidean distance measures. Their learning properties enabling the networks to devise the features needed for classification. Hence neural networks, with their inputs driven by the vectors to be classified and their outputs representing the classes, have been used in many situations where mathematical equations representing the classification have not been easy to find.

These properties are the reason for neural networks being applied in this research into facial image processing. As has already been discussed in Section 1.1, facial images

contain much complex information. It is one of the principle objectives of this work to investigate, and demonstrate, the extent to which ANNs are able to extract relevant information from face images in order to perform the task of facial feature classification.

1.4 Thesis Outline

This thesis describes the work performed by the author in order to investigate the possible solution of the problem of facial feature extraction using ANNs. Chapters 2 and 3 cover the main background areas to the work and the remaining chapters present the author's contribution to this field of research. The thesis follows the structure given below.

1.4.1 Chapter 2 - Introduction and background to facial image processing

Chapter 2 gives an overview of the history of facial image processing. It describes the major techniques that have been used and covers some of the applications of facial image processing including the use in police work and the history of police "mugshot" photography.

1.4.2 Chapter 3 - Introduction and background to neural networks

Chapter 3 is an introduction to artificial neural networks which covers all the techniques that are used in this work. Included in it is a description of traditional computing systems and biological neural systems to explain how artificial neural networks fit into the computing framework and where the original inspiration for them came from. Mathematical derivations are given for the operation and learning algorithms of each of the networks along with description of typical uses of each network type.

1.4.3 Chapter 4 - Data and data pre-processing

Chapter 4 discusses the data set used for this work, how it was gathered and the previous uses of the data. The data consists of 1000 face images and an associated

description which, although consisting of a number of numeric values, is designed to be meaningful to a human observer. The aim of this work is to investigate the automatic extraction of these descriptive measures from the images using artificial neural network techniques. The raw data set is not suitable for presentation to artificial neural networks so a number of different pre-processing techniques were used. Since the images were supplied in colour, it was necessary to explore different digital colour representations to find a suitable form for network training. In addition, the data was analysed using some statistical and un-supervised neural network methods to evaluate the separability of the classes involved.

1.4.4 Chapter 5 - Evaluating neural techniques using a simple classification

Chapter 5 describes the work done in applying artificial neural networks to the classification of simple facial features. The moustache and beard were taken as test cases to evaluate the potential for further development of these techniques. These are both cases of detecting the presence or absence of a feature rather than applying a classification to a feature that is always present. In order to provide a reference outside of neural network methods, consideration is also given to the use of linear regression and distance measures as alternative classifiers. The details of the methods employed are presented along with the corresponding results.

1.4.5 Chapter 6 - Putting neural techniques to the test with complex classification

Chapter 6 is concerned with neural network classification of more complex features. The facial area of the eye was chosen for this work and neural networks were applied to the classification of the eye colour. The discussion outlines some of the difficulties associated with this particular classification. Statistical methods are used for comparison with the neural techniques along with futher discussion of the method used where it differs from that used in the previous chapter.

1.4.6 Chapter 7 - Conclusions and future work

Chapter 7 draws together the conclusions that may be made from this work and suggests further work that may be considered in this area.

Appendix A contains the mathematical derivation of the derivative of two activation functions that are commonly used in neural networks. See Chapter 3 for the use of these functions and their derivatives.

Appendix B contains algorithms for the conversion of RGB colour values to the CMY and HSV colour representations.

Appendix C contains a paper published as a result of some of the work related to that contained in this thesis.

Chapter 2

Face Image Processing

Many studies have been undertaken on issues relating to the human face. It is clear from observation that faces are similar, with the basic features following a common form for most examples and yet it is possible to distinguish between many different faces and humans have seemingly amazing abilities to recognise known faces. In this chapter, an overview is presented of the main pieces of research work that have been done on facial image processing along with a summary of the police's interests in facial images. The subject of facial recognition has been the area in which most work has been performed and will therefore receive the most attention in this chapter.

The area of classifying facial features as performed in the experimental work of this thesis has received very little attention. That which exists is mentioned in this chapter, however it is generally based around conventional image processing techniques rather than the ANNs which are the focus of this thesis.

2.1 Conventional facial recognition techniques

The problem of facial recognition has been studied since at least the 1960's with various attempts to understand this most complicated and yet common of human abilities. Some research has looked specifically at what information is used by humans in performing facial recognition, and the degree of accuracy that can be achieved; while more recent studies have usually centred on computerised techniques for such recognition.

2.1.1 The human ability to recognise faces

Investigations have been carried out by various researchers into the human ability to recognise faces, usually focusing on the various psychological aspects of the process. A summary of the findings of this research will now be presented.

Initial studies by Goldstein *et al.*[24] were performed using a group of jurors to produce descriptions of a set of 255 photographs of faces based on 34 features. Various statistical analyses were performed on the descriptions to evaluate which were giving the most information and which were the most consistent between the jurors. This process reduced the set to a final list of 22 features. The recognition process was then investigated using a binary search technique where each feature of the face to be matched was taken in turn and the set of target images ranked from 0 to 1 according to the degree of match in this feature. Only the "extreme"[1] features of the target face were used and the features were matched in the order of the most extreme first. A "cutoff" criterion or threshold, p, was used such that the set was split at the point p or $1-p$ depending on the match being made and thus the target group was reduced. Investigation was made to identify the number of decisions that were required to reduce the set to a single photograph depending on the number in the population being searched and the value of p. A logarithmic relationship

$$r = -\frac{\ln N + \ln 2.62}{\ln p}, \tag{2.1}$$

was found between the population size and the number of extreme features needed for identification, where N is the population size and r the number of features needed. Thus giving an estimate of the number of extreme features needed to uniquely identify a face from within a given population.

Harmon followed up this work[29] by looking at the amount of information needed in an image to make a face recognisable. He tried reducing the resolution of the images by sub-sampling and then tested the resultant images on a group of subjects to evaluate the recognition rate. The results from these experiments showed a great variation in

[1] "Extreme" features were said to be those with a rating that differed greatly from the mean value.

the recognition rate from one image to another when the images were reduced to 16×16 pixels. As part of the investigation as to why this occurred, the images were re-sampled with the centres of the sub-sampled pixels moved right or down or both by half a block. This yielded four different versions of each face and it became apparent that each of these had a different recognition rate. Using the optimum image for each of the faces doubled the recognition rate achieved to 95%. The means by which these "block portraits"[2] hide data was then investigated with the conclusion that they introduce high frequency noise. This is why such images are more easily recognised if one looks at them with a squint since doing this effectively performs a low pass filter on the image.

Harmon also performed further experiments[29] on the recognition process using both a binary style search such as Goldstein's, already described, and a rank based system which allows occasional mistakes to be made whilst still resulting in the identification of the correct target. The binary sort process runs the risk of eliminating the target photograph early on in the search process by removal due to a mistake in the classification decisions. With the ranking system, at no time are any of the photographs removed from the data set, rather, at each ranking operation, a weighting is applied to each photograph based on how well the description of the feature in question is matched, thus allowing an image to still be matched with the target even if one of the feature descriptions is not correctly matched.

When the binary sort was performed by a human judge, on average 7.3 sorting operations were needed to isolate a single photograph whereas with a computer performing the sorting, only six decisions were needed. In addition to replacing the binary search with a ranking process, the searching procedure was altered in these experiments in an attempt to minimise the number of features needed to identify the correct face and to improve the accuracy of that identification.

As the feature descriptions were entered into the search, the user of the system was asked to input first the most distinctive features of the face being described. Once all

[2]The name "block portraits" was used to describe the way in which, when image resolution is reduced, the facial images appear to be made up from a number of blocks rather than being smooth pictures.

of these "extreme" features had been entered, an automatic feature selection process was used such that the computer asked for the "most discriminating feature" at any stage of the search. This was done by examining the faces in the set being considered to evaluate which feature had the most even distribution of values. For example, if all members of the population had small eyes then a decision based upon that would not work well for discriminating. However if there was a broad range of the size of mouth then that would form a good basis for a decision. In the system implemented, the computer always asked for 10 features as equation (2.1) suggested that this would be sufficient to identify the required target. This approach yielded accurate results with the target face ranked in first place for 70% of the trials and within the top 4% of the population for 99% of the trials.

Various psychologists have proposed theories of how the human visual[3] and recognition[9] systems work. Baron[3] examined theories of the human visual system with particular reference to the problem of human facial recognition. He devised a number of computer based and neural network models to evaluate these theories. Use was made of a technique in which low resolution copies of the faces, and feature areas extracted from the faces, were stored as templates against which new images could be compared. He described neural network models for standardisation of image size and considered the transformation performed on the image by the visual cortex. Lastly he examined the effect of damage to the network systems that he proposed, showing similar behaviour to damaged human systems. From these experiments, some insight is gained into the methods by which the human visual system is able to recognise faces. The details of exactly how the human visual system works is still well beyond our current understanding, but what knowledge has been gained may be of use in designing automated systems for face recognition.

2.1.2 Surveys of automated face recognition

A number of surveys have already been produced looking at conventional image processing techniques involved in facial recognition[7, 10, 70]. The main conclusions of these follow.

Samal and Iyengar[70] looked at five areas of the problem of facial recognition:

- Representation of faces
- Detection of faces
- Identification/Recognition of faces
- Analysis of facial expressions
- Classification based on physical features

Consideration was given to the amount of data needed to represent the image of a face such that it may still be recognisable. It was concluded that greyscale images at a size of $32 \times 32 \times 4$bpp[3] may be sufficient for identification purposes while it is certainly enough for detection of a face within an image. This may be compared to the images that were used as the basis of the work in this thesis. As described in Chapter 4, they are colour images at a size of $384 \times 512 \times 24$bpp when viewed as complete face images. The images used in this thesis, therefore, are much larger than is needed to identify the individual in the image. However, that is not the purpose of the work. This thesis is concerned with the detail of the features in the image and therefore requires images much larger than that needed for overall facial identification.

Samal and Iyengar[70] also noted many assumptions that are usually used in the facial recognition process such as the face being at a known position with no occlusion or rotation. Frequently samples are used with no facial hair or glasses and often the population is restricted to white males within a certain age range. Bearing these limitations in mind, it would appear that many of the attempts at designing facial recognition systems fall far short of what may be needed in a real life working situation.

Throughout most of the various work on face image processing, faces were represented in two basic forms, either using 2D intensity maps (i.e. the grey scale or colour image) or feature vectors (i.e. where individual features within the image are represented collectively as a vector, e.g. a vector denoting the outline of the an eye). The intensity map is the approach used in the template matching described in Section 2.1.3.

[3]bpp - bits per pixel

Its main drawback is that it generally requires comparatively large storage for each face, although if the minimum size of 32×32×4 already discussed is used, then the storage requirement is 512 bytes per face. In comparison, the images used for the work in this thesis are large at 589,824 bytes in size. However, this full size image is only being used as the source from which descriptive measures are to be calculated and will not be stored after that. See Chapter 4 for details on the data used in this thesis and the descriptive measures evaluated.

Feature vectors are derived from two basic sources - either the intensity map or a face profile. The first type are typically measures such as size of eyes, distance between eyes, size of mouth etc. Features from profile images are obtained from a set of points on the profile from which various distances and angles are measured (see Section 2.1.5 for some work in this area).

The detection and location of a face within an image, possibly containing a cluttered background, is an area in which much less work has been done as most recognition systems assume that the face is in the image within certain bounds. A system has been developed by Govindara *et al.* for locating faces within newspaper photographs[26], making use of the caption information to determine the number of faces that are present. This system matches against a simple model of a face outline consisting of two vertical straight lines and two arcs. The faces must be upright and unoccluded within the image for this approach to work. Others such as Craw *et al.*[13] have matched against more complex templates, using a multi-resolution method, starting with coarse images and moving to finer detail to give the accurate location.

Alternatively, the face image is located by finding some component part of the face and then deriving the overall face location from this information. This moves into the area of feature location such as is discussed on page 20 using deformable templates and other techniques such as the Hough transform.

Having established that a face is present in a given image, attention is now turned to the problem of face identification. This task is usually solved using a defined set of independent features which are matched against a database of the known faces. Various methods for doing this will be described in the following section, in terms of

processing either *full face* images or face *profiles*. The matching process is performed by a number of different techniques, often based on Euclidean distance, clustering, set partitioning or correlation methods. The precise technique used will depend on the features involved in the matching system; for example, a grey scale matching process may well be suited to correlation techniques whereas those representations with small vectors may be better served by a Euclidean distance measure.

2.1.3 Template and feature-based matching systems

As mentioned in Section 2.1.2, two basic methods are generally used for automated face recognition. That is, using either sections of face image, possibly the whole face, or measures taken from the individual features that make up the face. The latter may be referred to as feature-based systems or feature vector systems. The term *template* has been used with different meaning by different authors. In some cases it refers to areas of the image that are used in the matching process, whereas in others it refers to a model that is to be fitted to a given feature within the face images.

Brunelli and Poggio[7] compared the merits of using template matching and feature-based matching in facial recognition. In this case, they use the term template matching to refer to the use of image segments for the matching process. The feature based system[8] that they used searches for the locations of various key features on the face such as the eyes and mouth and uses these as an index into the database of faces to be matched against. The template matching system that they implemented was based on comparisons of grey level images of faces. The images are first *normalised* by making use of the positions of the eyes; the images are rotated, re-scaled and shifted such that the eye centres appear at fixed positions in all the images. This is done to eliminate variation in a face image due to position of the face within the image and distance of the face from the camera.

Eye locations were detected with templates that were correlated with the image such that the points of correlation where the greatest responses were obtained were taken to be the positions of the eyes. To reduce the computational load of the many correlation operations that would be needed to perform this search, a hierarchical sys-

tem was implemented, starting with a low resolution version of the image to identify the 'correctness' of search.

Four different schemes were then adopted to overcome variations due to illumination, including scaling of the image intensity[4] and use of the intensity gradient[5]. The comparisons were then performed using a cross-correlation function, in which the unknown image was compared with each stored image in the database. The work showed that template matching yielded a higher recognition accuracy (around 97%, compared with 90% for the feature vector) and used a simpler method. However the feature matching approach gave the possibility for higher speed of operation and a reduced memory requirement since the data for each face was held within 35 bytes compared with typically several hundred bytes needed for the templates.

In Brunelli and Poggio's work[7], various techniques were used for feature extraction including the following. Integral projection was used in order to locate both the nose and the mouth. In this process the pixel values are summed over an area in either the vertical or horizontal direction and the shape of the resulting histogram is used to detect the presence of certain features. The vertical integral projection is defined as

$$V(x) = \sum_{y=y_1}^{y_2} I(x,y) \ , \tag{2.2}$$

where $I(x,y)$ is the image and y_1 and y_2 are the vertical limits over which the integration is to be performed. For example in the detection of the nose, the absence / presence of peaks in the vertical projection is used to delimit the nose horizontally.

Another technique used by Brunelli and Poggio for feature location employs the gradient information contained within the grey scale image. For example, using the vertical gradient map, the eyebrows are found by looking for two peaks in the gradient information in opposite directions; the search area being limited to just above the eye position. An example of this using data taken from the data sets used in this thesis is shown in Figure 2.1. This shows the gradient values of a set of pixels on a straight line running vertically through the centre of one of the eyes. Also shown is the grey-scale

[4]generating pixel values as the ratio of the local value over the average brightness in a suitable neighborhood

[5]pixel values being replaced by the magnitude of the gradient in the image brightness at that point

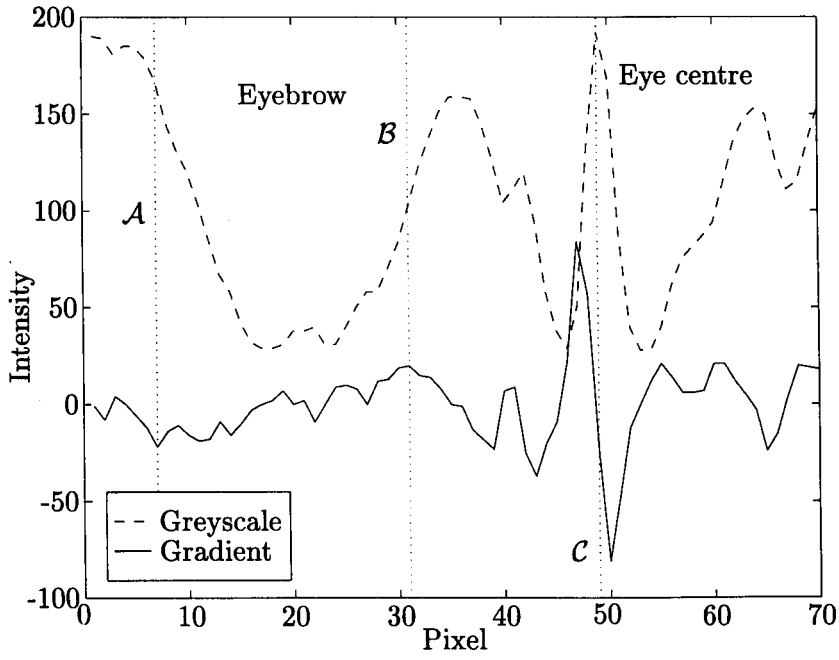


Figure 2.1 Gradient intensity used to locate eyebrows

values from which this gradient information was derived. The right most vertical dotted line, $\mathcal{C}$, denotes the position of the centre of the eye with the peak in the grey-scale corresponding to a white reflection in the pupil of the eye. The other two vertical dotted lines, $\mathcal{A}$ and $\mathcal{B}$, show the two peaks in the gradient, one positive and one negative that would identify the position of the eyebrow in this instance. The left hand side of the plot corresponds to the top of the image. In the system implemented by Brunelli and Poggio, these 'pairs of peaks' are compared for the two eyes and the most similar in terms of distance from the eye centre and thickness are chosen as the correct pair.

The face outline is found by following the outline on a gradient intensity map of the face projected onto an elliptical coordinate system. An elliptical coordinate system is used since the outline of the face is essentially elliptical and so a typical face outline becomes a straight line. The outline is taken over part of an ellipse covering the lower part of the face from ear to ear and eleven radii are used to describe shape of the chin.

Both of these methods used by Brunelli and Pogggio for facial comparisons start by locating the positions of the eyes. This is performed by template matching using a set

of five templates of differing sizes taken from images of one of the authors' eyes. The face images were then scaled by using the size ratio of the template which returned the greatest correlation with the face in question. That is, for example, if the template of an eye scaled to 0.85 times the original size gave the greatest correlation, then the face image was increased in size by a ratio of 0.85:1. Following this, the positions of the eyes were refined using left and right eye templates and rotation is compensated for by requiring that the eyes be on a horizontal line. The final step was to re-scale the image to give a fixed inter-ocular distance. This approach has proved to give a reliable method of normalisation.

The process of normalisation highlights an important area in the processing of facial images - that is the scale of the images needs to be consistent for the subsequent work to have any meaning. In a case like this where faces are to be matched, there is an obvious need for the techniques being used to be independent of the scale of the images. The same is true for the process of extracting a description of a face from an image. If the description is to include measures based on the physical aspects of the face, then it is important that the size of the images is consistent otherwise the descriptions based on these measures will have no meaning due to variation in the image scale. However, the feature classification performed in this thesis was done without any scaling of the images since the photographs were taken under controlled conditions resulting in identical scaling in the images. See Chapter 4 for details on the data used in this thesis.

Huang and Chen[36] used two methods for the location of facial features. The first approach, the deformable template, makes use of *a priori* knowledge about the shape of a feature to guide the contour deformation process. The template consists of an outline shape for the feature in question which may be deformed by altering a number of parameters. The deformation is an energy minimisation process which works on an energy function linked to suitable features such as peaks and valleys in the image intensity. The deformable template approach is advantageous compared with traditional edge detection routines in that it takes a more global view of the problem. The *a priori* knowledge of the shape of the object being located reduces the possibility

of the deformable template identifying the wrong region of the image as compared with edge detection routines which do not inherently employ this "knowledge" of the object they are locating.

The second technique used by Huang and Chen was the *active contour model* or *snake* which is an energy-minimising spline guided by external constraint forces and influenced by "image forces" such as lines and edges. These do not require their targets to be constrained in their shape as much as the deformable templates do, and are thus suited to extracting features such as eyebrows or face outline that are not so consistent in shape from one face to another.

In their work, Huang and Chen precede both the deformable template routine and the active contour by a rough estimation of the desired position of the features. This is done using what Huang and Chen call their "Rough Contour Estimation Routine" (RCER) which makes use of geometric presumptions to find a starting point for the estimation. For example in the location of the left eyebrow, the RCER presumes that its position is about 1/4 facial width.

While their work[36] is aimed at the recognition process, only the feature extraction is mentioned in their paper. No detail is given as to how the system could be implemented as a working recognition system. It is the view of this author that the recognition process would involve the matching of the extracted features; although with the active contours in their current form there would be considerable data to be stored and compared.

The deformable template model was also used by Yuille *et al.*[90] in order to locate the eyes and mouth. In their work, more details are given concerning the operation of the deformable template models. The templates act on the image and three other representations of the image which highlight such features as the edges, peaks and valleys within the image. It is these other versions of the image that are used by the template to locate the features within the image. The deformable templates are defined in terms of the peaks, valleys etc. within the image that they should lock onto in order to find the desired feature. Eye position location was found to be reliable provided the search starts in or below the eye.

Two different templates were used in the mouth position location, one for an open mouth and one for a closed mouth. In both the eye and mouth location algorithms, the search was implemented over a number of iterations, where the search coefficients linking the template to the valleys and peaks were varied as the search progressed. If the search starts above the eye, then problems are encountered with the peaks which are often found around the eyebrows. Their work was extended to perform tracking of an eye through a series of frames which gives good results provided that the eye does not move significantly between frames. This was probably due to the fact that they were using the template position from frame n as the starting point for frame $n+1$ and as has been mentioned above, the template only performs well when its initial position is in the 'region' of the feature that is to be located.

In this context, it would appear that deformable templates could be defined for other features **within** the face outline such as the nose. What is less certain is to what extent the principle could be generalised for "external" features such as the ears or hairline. Another aspect of the work by Yuille *et al.* that needs consideration is concerned with the need for some mechanism to permit interaction between the templates for the various features such that their geometrical relationship to each other is considered when evaluating a possible positioning of the template. To date, this author has not found any work in these areas.

In "real world" applications of recognition systems, the alignment of a face with a camera is difficult to control precisely. For this reason, there is a need to develop rotation invariant systems for facial recognition. To achieve rotation invariant recognition, Beymer and Poggio[4] use a series of 15 views of any given face as templates for matching against the unknown face in a system that matches using full facial images. Their work compares the results achieved when using 15 "real" views, taken with a camera, with the use of a set of views where only one of the views originated from the camera and the other 14 are derived from that one. This derivation is performed using a prototype face for which all 15 views are known and thereby the translations of areas of the image between views may be evaluated. These translations may then be applied to the single image from the camera and used to generate the "virtual" views

of the real face. In the comparison of recognition rates, the use of the 15 real views gave a recognition rate of 98% whereas the virtual views achieved only 85%. This is to be compared to the case of using one view and its mirror reflection where a rate of 70% was achieved. The high recognition rate when only camera based images are used might be expected since this is simply a case of finding an exact match within the data base. A recognition rate of 85% using the virtual faces shows that the technique used for producing these images yields a good representation of the face when rotated in the manner described and shows significant improvement over the use of a single view.

2.1.4 The Karhunen-Loève transform and "Eigenfaces"

Digitised images of faces often have large storage requirements, for example a 256×256 pixel image with each pixel representing one of a possible 256 grey levels will need 64Kb of storage if no compression technique is used on the image. While this is not a vast amount of space if only a few images are to be stored, the requirements can become very large if the database consists of thousands of face images. Standard image compression techniques such as those used in the GIF, TIFF or JPEG image formats may be used to reduce the image size but better compression may be achieved if a technique is devised that uses *a priori* knowledge of the problem.

The Karhunen-Loève transform, otherwise known as principal components analysis (PCA)[38] (see Section 4.4.1 for work using PCA in this thesis), has been used by Sirovich and Kirby[78, 39] as a data compression method for the storage of face images. The principal components of an ensemble of faces were calculated and used as the basis of an *optimal* co-ordinate system on which faces from both within and outside the original ensemble could be mapped. Investigation revealed that 50 principal components produced a good likeness of faces that were not part of the set used to form the principal components, giving around a 100:1 reduction in the data stored from the original images.

Moghaddam and Pentland[55] use the Karhunen-Loève transform as a means of locating faces and face features within an image and then encoding the face, including scale and translation information, into a total of 105 bytes of data. The feature location

process, using the Karhunen-Loève transform, is an alternative to the standard template matching procedures that are used to locate a given object within an image. It consists of calculating the "distance-from-face-space" of each pixel in the image to determine the "faceness" of any given pixel and thereby locate the face. This scheme allows for a greater distortion in the object that is to be detected and has been shown to give superior performance to matched filtering[41]. This system is intended as a possible compression method for use in video telephony or some similar application where high levels of compression are needed. It's effectiveness is demonstrated in [55] by comparison with a JPEG image at the lowest quality[6]. This yields a 540 byte image which is barely recognisable as a face, whereas the 85 byte Karhunen-Loève representation is a close match to the original.

Turk and Pentland[81] used the same method of principal components analysis and gave it the name "eigenfaces". The method of eigenface encoding may be thought of as a form of feature based coding of the facial image. However, in this approach, the features are not individual sections of the face such as eyes or nose, but rather are the principal components of a collection of images; thus the features have little meaning to the human observer. In their work, Turk and Pentland looked at the application of eigenfaces for recognition, using the eigenface method to reduce the storage requirements and computation requirements in comparing a new face to those stored in the database of known faces. In addition to reducing the data storage and computation requirements of the matching process, the use of eigenfaces transforms the images into an optimal co-ordinate system for the representation of faces and therefore improves the matching accuracy and aids the detection of images that do not contain faces.

Moghaddam and Pentland[54] have further investigated the use of the eigenface representation for facial recognition, using the eigenfaces to detect rotation in the image about the y-axis, thus producing a rotation invariant recognition system. Their work is a good demonstration of a practical system that yields high accuracies of 95% recog-

[6] A JPEG image is created using a compression technique that involves loss of the original data. When generating a JPEG image, there is a "quality" parameter which determines the amount of compression applied to the image. The more compression applied, the greater the loss of data from the original

nition rate using a database of approximately 7500 photographs taken of around 3000 people with a mix of age and ethnic group. Further, using "eigenfeatures" they have produced a method of facial feature location yielding a 94% detection rate compared to a more traditional sum-of-square-differences matching method which gave only 75%. Then, using these eigenfeatures to code the face images, they established a method of facial recognition that showed a greater invariance to distortions in the image than the whole face, eigenface method. The combined use of the whole face and the eigenfeatures resulted in a 98% accuracy in recognition.

Jiang[37] has combined eigenfaces with neural networks in his attempt at a face recognition system. The eigenface representation of faces are used as the input to an artificial neural system that has been trained to identify the face. The face images are represented by the first 100 eigenvalues, dramatically reducing the data size from the original 46×46 pixel image. While the reported recognition rates are high at more than 95%, the database size used in this particular work is small, i.e. only six different people, and therefore the technique needs to be tested in terms of scalability up to the kind of database sizes that may be seen in "real world" applications. See Section 2.3 for further information on face image processing using artificial neural networks.

2.1.5 Use of profile face images

Thus far, with the exception of the systems that have attempted to be rotation invariant, all the face recognition systems presented have used full face images. Other possibilities are available and some researchers have investigated them.

As one example of this type of work, Wu and Huang [88] looked at the use of a profile of a face and defined six points on that profile curve from which 24 features are extracted. The six points on the profile image, as shown in Figure 2.2, are located by a variety of techniques. The tip of the nose is assumed to be the extreme right hand point (face looking to the right), the bottom of the nose and the mouth points are established by the use of cubic B-splines, the chin and forehead points are determined by measuring distances between the other points already found and then mirroring these distances. The eye point is found by taking the point on the curve furthest from

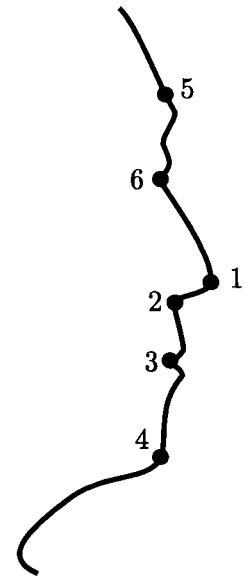


Figure 2.2 Profile points used by Wu and Huang

the straight line between forehead and the nose.

From these six "points of interest", five features were calculated for each of the five curve segments connecting the points. These were:

- Distance between points (as a straight line.)
- Angle between curve segment and the neighbouring curve segment.
- Length of curve segment.
- Summation of curvature values at each point.
- Symmetrical measure of the curve segment.

The recognition results from this system appear to be excellent in that it achieved nearly 100% accuracy. Even so, this approach may be of little practical relevance since the context of the work is too far removed from the type of system that could be used in a practical application. For example in this system, a method of back-lighting the head was used to get the image of the head, leading to a very contrived setup that bears little relationship to the types of situations where recognition is likely to be needed.

They also only used a very limited data set of eighteen people which does not reflect the number of subjects likely to be found in a real life situation.

2.1.6 Face expressions

Work has also been undertaken to analyse facial expressions and hence to see how these will affect the recognition rate of a face recognition system. Recent work by Yacoob *et al.*[89] used sequences of video of 130 people showing "no-expression" and a selection of expressions from the six "universal expressions"[7]. Two different face recognition procedures were used; eigenfaces as proposed by Turk and Pentland[81] and feature-graphs[47]. Results were presented showing the distance of the closest three matches in the database to the image in each frame of the sequence under consideration. The work indicated that the eigenface method would prove to have a greater immunity to the expression being shown on a subject's face when attempting to recognise that face. In these video sequences there was a greater distance between the correctly classified face and the next closest one when using eigenfaces and this distance was not greatly affected by the expression on the face.

Some investigation was also made by Yacoob *et al.* into the role of gender which revealed better distinctiveness within the database of female subjects. This was thought to be due to the greater variation in hair style and use of makeup. However this result may not be as clear cut as it is presented since the number of female faces in the database is only 23 whereas the number of male faces was 103 thus making the comparison statistically biased.

2.2 Police uses of face images

Photographs are popularly perceived as an important part of police work with the use of "mug-shots" and other face likenesses in "wanted" posters and the like. The history of face photograph usage is, not however, one of the efficiency and effectiveness that might be expected from the police. There have been many attempts at standardisation which have only been partially implemented, or indeed not at all. The primary use of

[7]These are anger, disgust, fear, happiness, sadness and surprise

face photographs within police work is that of identification of suspects. Making use of the human ability to remember and recognise faces, the police may use photographs of previous offenders as a means by which a witness may identify a suspect. Once a suspect has been identified, photographs of that person may be released to the public as part of a means of tracking down the suspect.

The earliest known police photographs date from 1843 [79], and these were produced by the *daguerreotype*[8] process. By 1865, police photographs of suspects were used on wanted posters but there was no standardisation in the process until 1884 when Alphonse Bertullion[64] used a pair of photographs, one full face and one profile on a single plate. This was then formally recognised in 1888 when he was made Prefect of the Department of Judicial Identity in Paris.

In the UK, there was little standardisation accross the regional police forces, each adopting their own methods of photography and combinations of different view points of the faces. Attempts to standardise started in 1947 when the Department Committee on Detective Work and Procedure produced recommendations for a three-view standard[9]. However differences still occurred and following another report in 1979, most forces adopted a two-shot system, leaving out the full length one, while the Metropolitan Police force chose to use a single three-quarter-profile shot. Other problems continued, mainly due to operator error and insufficient training. In 1988 it was reported by Inspector Cox[12] that, of the prisoner's photographs taken in Hertfordshire. 20% failed to produce an image and a further 30% were of poor quality.

Another aspect of police use of facial images is the production of face images from a witnesses description. This is performed by one of two methods; either using an artist's sketch or by use of one of the systems for constructing a face from a series of components (e.g. the 'Photo-fit'[10] kit.) Both systems use trained operators to produce

[8] An early photographic process in which the impression was taken upon a silver plate sensitised by iodine and then developed by vapour of mercury. (Shorter Oxford Dictionary)

[9] The committee recommendation stated that the following three views should be used:

- Full length with hat on, $\frac{1}{27}$th natural size.
- Head and shoulders, full face, looking square into the camera without hat and showing the set of the shoulders, about $\frac{1}{8}$th natural size
- Head and shoulders profile, right side of the face, again about $\frac{1}{8}$th natural size

[10] The Photo-fit kit was devised by Jacques Penry and is marketed by John Waddington of Kirstall

the images. The use of an artist's sketch requires a skilled artist who interviews the witness to obtain a description of the suspect and uses that information to draw a 'likeness'. The witness will be present while the sketch is being made and will make suggestions as to what needs changing to improve the likeness. The Photo-fit method also requires a skilled interviewer to obtain the witness description and construct the face likeness. Again the witness is present to enable the likeness to be altered as it is constructed.

Ellis *et al.*[18] made an investigation into the effectiveness of the Photo-fit system for reconstructing face images. They used both real photographs and Photo-fit produced faces as the targets for a set of subjects to attempt to re-create. The conclusion of their work was that it is in fact a difficult task to accurately reconstruct a face using the Photo-fit kit, even with the original visible, due to the difficulty in choosing individual features within the face rather than considering the face as a whole. In the 64 trials that they performed, the five features were correctly selected as follows: forehead, 45 times; eyes, 28 times; nose, 9 times; mouth, 26 times; chin, 5 times. When examining the recognition rate that was achieved, using the Photo-fit reconstructions as stimuli, Ellis *et al.* showed by various tests that likenesses produced by "good" encoders[11] were more easily recognised than those produced by "bad" encoders. Although detailed numerical results are not given, statistical tests were performed to prove the significance of the difference between the two groups thus demonstrating that the accuracy of the likeness produced by a witness will greatly affect its use in recognition of the suspect.

Further work by the same group[17] investigated the effect of time of exposure to the target face before reconstruction, skill of the operator of the Photo-fit kit and the difference between working from memory or with the target face visible. In each of the experiments, a set of face likenesses were produced by a group of subjects and were then judged on a seven point scale according to their similarity to the target face. They found that there was no significant difference between the quality of face reconstruction due to the variation in time of exposure to the target photograph. There was also no

Ltd.

[11]The "good" encoders were those whose likeness to the Photo-fit targets used the greatest number of correct parts from the kit and were deemed to be a good likeness

significant difference in the likenesses produced under the guidance of a police operator, compared to those produced by a research operator. In comparing the results achieved with the target faces present with those when it was absent, in addition to Photo-fit likenesses, subjects were asked to sketch the target face. This experiment showed that the sketching operation was significantly affected by the presence of the target face. In this case the likenesses were scored out of 100 and the mean score achieved with the target present was 56 whereas without it the score was only 14. However, reconstruction using the Photo-fit likenesses showed little difference in the quality of the image produced between the two cases. The small variations observed in the scores for the Photo-fit likenesses under these varying conditions would indicate that there is a fundemental limit to the accuracy of likeness that can be generated with the system. This could be a problem that needs addressing in the construction of likenesses by the police.

An alternative system for helping witnesses to recall faces is the 'Identikit'[12] system which was a forerunner of Photo-fit. Whereas Photo-fit is based on photographic representations of the various features, Identikit was designed using line drawings of features printed on sheets of acetate. A study on Identikit was performed by Laughery[44] in comparison with likenesses produced by sketch artists. A set of 71 white males were used as target faces and 142 "witnesses" were used to produce the Identikit and sketch likenesses of the targets. A series of three Identikit operators and three artists produced the images from the witnesses descriptions. This concluded that sketch artists produced better results than those from the Identikit system, mainly because of the greater flexibility of the artist who is not constrained to a fixed set of features. Several of these evaluations of face recall systems are summarised by Davies[14] who also considers the psychological elements involved in designing such systems. He describes the then (1981) future possibility of computer based systems that would allow much greater flexibility in the manipulation of facial feature elements, such as stretching and shortening.

Having described the two main areas of facial image work that the police primarily

[12]The Identikit is manufactured by the Identi-Kit Company, 17985 Sky Park Circle, Suite C, California 92714, USA.

employed consideration will now be given to a third and more recent area. That is, how can the "mug shots" of criminals kept on file be used in helping to solve crimes? The primary use of these mug shots is to assist witnesses in their attempts to identify suspects. However there is an obvious problem here in that the witnesses have to examine huge databases of photographs; hence the success rate in identification is low, police forces reporting rates as low as 2%[79]. For this reason, various researchers have attempted to develop computerised techniques to assist with the process of searching a database of images.

Work on this type of system began in the mid 1970's[45] and is the area which forms the focus for the work undertaken by the current author. The FRAME[13] system[73] was one such example of a solution to the problem. It made use of a set of 1000 faces which were stored on video disk along with an associated database of 50 coded attributes per face. These 50 attributes are descriptions of the face in a "human understandable" form such as *size of eyes* or *length of hair*. Each is rated on a scale of 1 to 5 (unless it is only appropriate to have a "yes/no" form of rating) and stored to an accuracy of one decimal place. These are the "descriptive measures" used by the author in this thesis, and they are described in more detail in Section 4.1.1.

When in operation, the witness' description of the target face is entered into the computer system by means of these descriptive measures and used to perform a search on the database. The twelve closest matching faces are then displayed on a video screen and if the target face is not amongst those presented, then the option is provided to alter the description as appropriate to obtain a better match. FRAME demonstrated that the system did indeed provide an effective means of retrieving faces from a large collection. It also showed that when the face to be identified was not a "distinctive" one, it was a significant improvement over the album method of searching where the witness simply looks through a large photo album of mug shots.

The FRAME system was followed up by FACES[14][1]. This system was intended to more closely reflect the operational needs of such a face retrieval system. It was specified to hold 20,000 face images and related descriptive data, with the possibility

[13]FRAME stands for Face Retrieval And Matching Equipment.
[14]FACES stands for Facial Analysis, Comparison and Elimination System

of being expanded to 100,000 faces. Due to the limitations of computer technology at the time, the computer interface was text only and the photographs held on microfiche using a frame store to provide display onto the screen. The changes from the FRAME system were mainly conversion of the experimental system into an operational one.

The evaluation of the FACES system continues[59] and it has been ported to the Microsoft Windows environment resulting in added functionality and ease of use. In addition an alternative searching algorithm has been implemented by making use of genetic algorithms[31] (GAs). The GA search method does not require the witness to provide values for the descriptive measures. Rather a selection of 10 dissimilar faces is presented and the witness is asked to order them according to similarity with the target face.

The GA search process uses pieces of data called "strings" consisting, in this case, of the facial feature values associated with the faces currently being displayed. The strings from the first set of pictures are "bred" to produce a new set of strings by the process of "crossover" and "mutation". Crossover is where a child string is obtained by combining parts from the parent strings and mutation is where elements in the child string are changed at random. This new set of strings is then used to search the faces database and retrieve a new set of images. In producing the new set of strings, the similarity ordering of the parent set is used to determine the likelihood of any given parent string contributing to the new child strings.

A comparison has been made[59] between face retrieval using this GA based search and the original version using the FACES codes which showed that the original codes outperformed the GA coding. It has been proposed that the reason for this was the limited data set used of 1000 males and that it is likely that for searching larger data sets, the use of the GA based search would be able to distinguish faces that the original method would fail on. Following on from these trials, a combined GA and facial feature search method has been proposed by Nicholls[59] though no results from this method have been published yet.

2.3 Neural network face processing techniques

Various attempts have been made to exploit the power of artificial neural networks (ANNs) in image processing applications (See Chapter 3 for background on neural networks). As is the case with conventional image processing of faces, the majority of reported work on facial image processing using ANNs has been concerned with the recognition of individuals, although there have also been various attempts made to locate feature points and classify facial expressions. What follows is a summary of some of the relevant work that has been performed, using neural networks, to process face images.

2.3.1 Recognition systems

The WISARD[15] system as described by Stonham[80] is a RAM based neural network which has been used to perform facial recognition on a set of 16 people in an unconstrained environment using one ANN for each person. The WISARD "learnt", in real time, each of the faces in a separate network using a series of 200–400 images taken from a video camera. During the testing stage, the images from the camera were presented to each of the 16 networks. The network which generated the highest response was assumed to indicate the correct identification of the individual concerned. This was shown to be successful for the case of the 16 class problem.

Although this performance is quite impressive, there are limitations to its use as one network is required for each face that needs to be recognised. In a practical system, there may be hundreds of faces that need recognition and this would require a corresponding number of networks with corresponding computational requirements. In addition, no mention is made here of the possibility that this method would be scalable to large problems - it is quite likely that with a large number of networks, the discrimination between faces would be low, causing a reduction from the 100% accuracy reported in [80]. Further consideration of the WISARD system is presented in 2.3.3

Bouattour *et al.*[5, 6] created a custom neural network architecture for facial recognition. The neural network they used made use of local connections, rather than being

[15]WISARD stands for WIlkie, Stonham and Aleksander's Recognition Device

the standard fully connected network, and shared weights which gave an order of magnitude reduction in the number of weights needed to implement the network. The architecture consists of four hidden layers which perform feature extraction. These features are not predetermined but are instead developed during the learning phase. The second part of the network acts as a classifier which combines the features extracted by the first part of the system, with each person represented as a separate class. Within the hidden layers, there exist cell clusters which are locally connected and use shared weights. These perform various filtering operations to extract the feature map of the image.

Their data set was less restrained than many others in that is contains examples of the subjects under different lighting conditions, with different orientations and with or without glasses and features which are required for practical applications. These additional variations are not usually considered in other systems because they are felt to be too "difficult". The system was however limited because only ten people were included in the data set and this obviously does not accurately reflect the likely population in a real system.

In terms of classification performance, the system gave around 89% accuracy when the search was performed on the ten people. The results improved when fewer people were included in the database, as one would expect. Consideration was also given to the performance achieved when "unknown" faces were presented to the network. The output cell activities were examined and different thresholds were tested to devise a compromise between non-recognition of a valid face and acceptance of an unknown face. The scheme used was based upon the examination of the values of the two largest outputs from the network and setting a decision threshold involving these. The results presented for this part of the work are unclear as no mention is made of whether the faces that were rejected were those that should have been recognised or whether they were "unknown" faces

Another neural network based approach to face recognition was proposed by Edelman *et al.* [16]. Their network was of the HyperBF type[61], a form of radial basis function network, where functions are approximated by the superposition of basis func-

tions of the form

$$f(x) = \sum_{\alpha=1}^{n} c_\alpha G\left(\| x_i - t_\alpha \|^2\right) , \tag{2.3}$$

where t_α and c_α are found during the learning phase. Their work used a database of 27 images for each of 16 people. Before presentation to the networks, the images were normalised by scaling such that the eyes and mouth were brought to fixed positions. The system then consisted of one HyperBF network trained to recognise each of the people in the database, using 17 of the images of the person in question for the training of the network. The 10 remaining images were used to test the network and the error rate achieved by the system was 22%. The system was then further expanded by the inclusion of a network trained to take the output of the 16 classifiers and produce the true classifications. This improved the error rate to 9%. Once more this system is limited in its practical application by the need for one network per person who is to be recognised. However, this system does use images taken under varying lighting conditions and with different camera locations so the data sets used are more in line with what might be expected in a "real" application.

Micheli-Tzanakou *et al.*[52] compare two different learning algorithms for the training of a feed-forward network for facial recognition. Their training data consisted of three different compressed versions of the digitised images. These were produced using F-CORE[16], developed by Micheli-Tzanakou and Binge[50], invariant moments[35] and wavelet decomposition. These three compression methods resulted in the original images being reduced to a set of between 8 and 13 variables which were presented to the network. They found that the ALOPEX routine converged quicker with a greater degree of accuracy than backpropagation though the results presented here are far from clear as to the precise experiments performed. ALOPEX is an optimisation process that was developed by Micheli-Tzanakou[51] and has been applied to a number of problems, in this case it was used in place of backpropagation for the adjustment of the weights of a neural network[82, 49]. If the problem to be optimised (in this case, training the network) may be defined in terms of a *response function* $R()$, then the parameters at

[16]F-CORE stands for Fourier based COmpression REconstruction

time τ, $X_i(\tau)$ are updated each iteration according to

$$X_i = X_i(\tau - 1) + \gamma \Delta X_i(\tau) \Delta X_i(\tau) \Delta R(\tau) + r_i(\tau) \ , \tag{2.4}$$

where γ is a scaling constant, $r_i(\tau)$ is a random number from a Gaussian distribution and $\Delta X_i(\tau)$ and $\Delta R(\tau)$ are found by

$$\Delta X_i(\tau) = X_i(\tau - 1) - X_i(\tau - 2) \ , \tag{2.5}$$

$$\Delta R(\tau) = R(\tau - 1) - R(\tau - 2) \ . \tag{2.6}$$

More recent work on face recognition includes that by Howell and Buxton who make use of RBFN[33] and their own variation on a RBFN[34] as methods of tackling the problems of position, rotation and scale invariant recognition (see Section 3.3.2 for a description of RBFN). In [33] 10 images of each of a set of 10 people were used where each image showed the face at a different rotation ranging from full face to profile. The images were sub-sampled at different resolutions and a 'difference of Gaussians' filter was convolved with the images followed by binarisation to yield the final image for training. Some of the work employed just two faces to evaluate the techniques used. The resulting strategy was then applied to all ten faces. Recognition rates as high as 78% were achieved with this system whereas a backpropogation system failed to converge for the same data set. The effect of shift and scale variance in the face images was also evaluated with best accuracy reported was now 37% when 1/5 of the images were used as training examples.

In [34] the work is extended to use a Gabor filtering technique as an alternative to the difference of Gaussians and the "face unit" network, their variation of a RBFN, is introduced. The latter is simply the case where one network is trained for each of the face classes to be recognised rather than the more conventional approach where a single network is used to perform the entire task. The Gabor filtering method was shown to result in higher recognition rates than the difference of Gaussians and the face unit network gave noticeable improvement in the results in respect of the effect of shift and scale in the images. Typical results indicate accuracies of about 80% depending on the

network architecture.

Sidney *et al.*[77] have adopted an optical neural network for facial recognition. The neural network is implemented in a holographic system which is trained by modifying the hologram pattern. During recognition, the "processing" in the system is performed by passing light through the hologram which therefore makes this a truly real-time system. However, once again the number of faces involved in the experiments is small; the system was trained to recognise one of the authors and was then tested against video of all three. The technique shows promise but needs to be tested for scalability to "real world" applications.

2.3.2 Feature point location systems

In addition to the recognition of faces, neural networks have been applied to the problem of feature point location within facial images. Vincent *et al.*[86, 85] use a multi-resolution approach to the problem, starting with a low resolution 16×16 representation of the image before refining the search in the 256×256 original image. A series of multi-layer perceptrons are used on the low resolution images to generate specific search areas for the features of interest. These search areas are validated by using spatial and temporal pruning algorithms which make use of *a priori* knowledge about the geometry of the problem. The temporal aspect is introduced since the system is intended to be used with real-time images produced from a video camera.

Having established the search areas for the feature points, the corresponding areas of the high resolution image are scanned to locate a total of twelve feature points; five for each eye and two for the mouth. When this technique was applied to their database most points were located within one pixel of their actual position and, with the exception of a couple of spurious images whose results had large errors, all points were found within four pixels of the true position. The database used by Vincent *et al.* is not described in detail, but from the pictures published, one may note that it consists of both male and female faces, with no restrictions applied to race or other features such as glasses and facial hair. The networks were trained using 30 of these images and a further 30 were used for testing along with a number of sequences of moving faces.

An RBFN based network was used by Debenham *et al.*[15] to locate eye position within an image. It employed the RCE (restricted coulomb energy) method to add new RBFN neurons as they are needed to represent the desired input-output mapping. The algorithm for this growth is given in Algorithm 2.1.

Algorithm 2.1 The RCE algorithm

```
for each input pattern do
  Apply a pattern on the inputs and calculate outputs based on
  this pattern
  for each output do
    if 'on' and should be 'on' then
      do nothing
    end if
    if 'off' and should be 'off' then
      do nothing
    end if
    if 'on' and should be 'off' then
      shrink the radius of all 'on' RBF neurons connected to this
      output just to touch the input point
    end if
    if 'off' and should be 'on' then
      'spawn' a new RBF neuron centred on this input point
    end if
  end for
end for
```

The searching process performed by these networks was restricted to a 64×64 sub-image surrounding the right eye, the centre of which was the desired target. The network was fed an 8×8 window of this sub-image; the window was raster scanned over the sub-image one pixel at a time. The system was trained on 30 images and tested on a further 30 and correctly located 28 of the eyes in the test data set. While these are good results, it should be remembered that these experiments were relying on the eye in question being within a given area of the original image so that the sub-image could be extracted. In many "real world" applications, it would not be possible to make such assumptions. The techniques used by Debenham *et al.* could be considered to be a method of locating the feature in question once a coarser method had established the general region of interest.

Herpers *et al.*[30] applied another variation of a RBFN, a so called Dynamic Cell Structures (DCS) network, to classify image positions detected by another search strat-

egy. The DCS learn the topological structure in the input space and were employed in this case to verify the results of model- and data-driven sequential search strategies in the location of a set of nine feature points around the eye. In addition, the effectiveness of the approach in respect of locating the position of the feature points was also investigated using the DCS network alone. This latter approach was shown to result in error rates of between 12% and 62% whereas the combined system of DCS and sequential search yielded errors of around 1% compared to the sequential search by itself which gave 17% error.

2.3.3 Expression classification systems

The WISARD[80] system, already presented in Section 2.3.1, has also been used in an expression classification system. The problem to which it was applied was a 'two class discrimination' between a smiling face and a serious face. This was tackled using one network per class, as was the case for the face recognition problem using the same system (see Section 2.3.1). In this case, however, only one person's face was used for the experiments. Various optimisation techniques were investigated for removing processing elements that were not involved in the classification. This seemed to have the unfortunate side effect of making the network problem specific rather than a general system. That is, this optimised version of the WISARD network in only suitable for performing the "smiling/serious" classification. To use a WISARD system for other classifications, it would require a different optimisation.

Rosenblum *et al.*[67] have made use of RBFN[57](see Section 3.3.2) in their system for identifying six "universal expressions", namely disgust, happiness, anger, sadness, fear and surprise. The method employed extracts motion information from a sequence of images, using the eyebrows and mouth as the features of interest. The motion information, in the form of "optical flow", is calculated using a correlation technique which assumes that the motion between two consecutive images is bounded within an $n \times n$ window, the size of which was determined for each expression. This motion information is then used as input to a series of six networks, one per expression. The input vector contains a feedback system which allows information from past image

frames to be incorporated into the network. This is done by adding the previous input vector, multiplied by a decay constant, to the current input. The resulting activation is defined as

$$C_i(\tau) = \begin{cases} 1 & \text{if } \alpha C_i(\tau - 1) + I_i(\tau) > 1 \\ \alpha C_i(\tau - 1) + I_i(\tau) & \text{otherwise} \end{cases} \tag{2.7}$$

where $C_i(\tau)$ is the activation of node i at time τ, α is the decay constant and $I_i(\tau)$ is the input to node i at time τ.

The database used by Rosenblum *et al.* consisted of 46 image sequences of 32 subjects showing the different expressions. The results are presented in terms of retention, extrapolation and rejection for a network responding to a given expression. That is, the network's ability to correctly identify expression sequences that it was trained with, the ability to correctly identify expressions from previously unseen sequences and the ability to reject expressions other than the one which the network was trained to recognise. The results given are 88% for retention, 73% for extrapolation and 79% for rejection.

Zhao *et al.*[91] have tackled the problem of classifying the same six expressions using a cascade-correlation neural network. Their input data, however, was a set of 10 measures taken by hand from a set of 94 photographs rather than any image representation and all the images were still photographs rather than sequences. A series of individual networks were trained for each of the expressions and a further network was used to resolve the outputs of these to a single labelled output. Accuracies achieved were high, ranging from 87.5% for 'sad' and 'angry' to 100% for 'surprised' and 'afraid'. The method used by Rosenblum *et al.* for this expression classification is fully automated whereas that presented by Zhao *et al.* requires manual intervention in taking the measures from the photographs. Hence the former method is more suited to practical applications where the desire is for as much automation as possible.

2.4 Colour image processing

At this point, it is worth considering some of the existing work in the area of colour image processing. Most image processing work is done using intensity maps, that is, the colour information is discarded as it is not required. However, in areas where the intensity maps have failed to produce adequate results, colour information has been considered.

Most frequently, colour is used as part of an image segmentation process. In their review of image segmentation techniques, Pal and Pal[60] discuss methods in which colour is used for segmentation. In their review, however, they do mention that:

> The literature is not so much rich on color image segmentation.

That work which does exist in this area generally uses one of two methods. In the first, sample pixels for a given area to be detected are known and distance measures in the relevant colour space are used to determine whether a given pixel is part of the area in question or not. Alternatively, where samples are not available through lack of *a priori* knowledge, clustering techniques are used to group pixels and determine image segments.

Aside from segmentation, Saber *et al.*[69] use colour information for classification of pixels into classes such as skin, sky or grass. They make use of an alternative colour representation called the *YES* colour space. This is calculated from the normal RGB representation by

$$\begin{bmatrix} Y \\ E \\ S \end{bmatrix} = \begin{bmatrix} 0.253 & 0.684 & 0.063 \\ 0.5 & -0.5 & 0 \\ 0.25 & 0.25 & -0.5 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{2.8}$$

For more information of colour representations see Section 4.2.1. Probability density functions of the pixel values in *YES* colour space for sample pixels are produced and when plotted on the ES axis, form elipses. The classification of new pixels is then a process of calculating distances from the position of the new pixel in the ES space to the centres of the elipses. A technique called "Maximum *a postiori* probability" is applied

to any pixels whose classification is ambiguous to produce a high level of accuracy. This paper is the closest conventional image processing example that the author found to the work presented in Chapter 6 on eye colour classification.

A face-related example of colour work is from Chen and Chiang[11]. In this work, colour classification of pixels by a neural network is used to identify areas of an image as being skin coloured. These areas are then manipulated to determine the position of faces within the image. This work has the obvious limitation of being trained to recognise oriental skin colour and does not mention examples of using Caucasian or African skin colouring.

2.5 Summary

Psychological experiments are helping to increase our understanding of the process which humans use when recognising faces. This understanding is enabling researchers to develop systems for facial recognition that utilise the same visual cues that are used by humans. In addition, this information is enabling police researchers to improve the search methods used in witness identification of suspects.

It is not practical to compare many of the different computerised face recognition methods that have been presented in this chapter in terms of performance because each uses its own data set and each is designed with a specific purpose in mind such as rotation invariance or scale invariance for example. However what may be said is that while many of the systems reported show good recognition accuracies, they use databases that are much smaller than that which will be found in real life applications. Much promise is shown in the use of the Karhunen-Loève transform, both for recognition systems and for compression of facial images, for storage or transmission as might be required in a video telephony system.

The neural networks community is applying a number of approaches to facial image processing. Many examples use conventional image processing techniques as pre-processing methods before presenting data to the neural networks. In some cases it is difficult to tell how much of the problem is being solved by the neural network and how much by the pre-processing.

As has already been stated, the majority of facial image processing work is in the area of recognition. The area of work investigated in this thesis, that of classifying facial features, is one that has received little attention and none so far from a neural network point of view.

Chapter 3

Artificial Neural Networks

Artificial Neural Networks (ANNs) have recently, over the last ten years or so, received much attention as a new tool for data processing. They are being used in many areas from control to image processing and are being presented by some as a solution to many problems. This chapter contains an overview of the history of ANNs and considers some of the key differences between ANNs and traditional computing techniques. The following definition summarises the properties of an artificial neural network:

> A Neural Network is an interconnected assembly of simple processing elements, *units* or *nodes*, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or *weights*, obtained by a process of adaptation to, or *learning* from, a set of training patterns[27].

3.1 The von-Neumann Machine

The traditional form of computer follows the steps of operation of a von-Neumann machine. This operates according to the following repeated sequence:

- fetch an instruction from memory
- fetch any data required by the instruction from memory
- execute the instruction (process the data)

- store results in memory
- repeat cycle

These machines have a pre-defined algorithm (the program) to work through and require that the problem to be solved is understood in an algorithmic way before this program can be written. Once the problem has been defined in this way, they are consistent in repeatedly performing the same tasks. However, if the problem to be solved is not able to be defined by an algorithm or a set of mathematical expressions then this method is of no use since there is no basis on which the program can be written. Von-Neumann machines also require that the data they are to process is precisely defined; any noise in the data can cause them to work incorrectly. They are easily subject to malfunctions in various ways; e.g. if only a small section of the "machine" is corrupted in some way, such as one memory location becoming corrupted, then the system can cease to work as expected or "crash". Finally, von-Neumann machines operate as serial machines; that is they performs a single operation at a time which can lead to slow operation of the system if the algorithm being followed is complex.

In contrast, ANNs are not as suceptible to the same set of restrictions as the conventional computer. They exhibit the following beneficial properties:

- Parallel structure can give high speed operation on specialist hardware
- Ability to generalise
- Resilient to noise
- Degrade gracefully when damaged
- Can solve non-liner problems
- Different network architectures to suit different problems.
- Supervised or unsupervised learning methods.
- May reduce development time compared with traditional computing methods.

Neural systems do not require an empirical solution to the problem but rather learn a solution by being presented with examples of the required responses. They are able to cope with noisy data in a similar way to that in which humans are able to recognise handwritten characters even from sources that have not previously been seen. Neural systems are also resilient to failures within the system. As the processing is carried out over a large number of elements and the information stored in the system is distributed across the network, the failure of some of the elements of the network does not cause the whole to stop working, rather there is a gradual degradation in performance as the damage to the network is increased. An artificial neural network is parallel in design and when implemented on appropriate hardware is a machine capable or performing many computations simultaneously.

It is these differences in behaviour that make artificial neural networks an area worth studying and a potentially useful tool in problem solving. However, in contrast with the von-Neumann machines, neural systems are not mathematically predictable in their behaviour, and, although a given trained network will always respond in a certain manner, another network trained on the same data is unlikely to produce exactly the same results. This property eliminates the use of neural networks in certain applications which are more suited by algebraic mathematical expressions.

3.2 Neuron Models

The basic element of all neural systems, whether biological or artificial, is the neuron. Next we shall consider the current understanding of the biological neuron which has been the inspiration behind the artificial systems. Following this, details will be given of some of the mathematical models that are in use in artificial neural systems.

3.2.1 The Biological Neuron

The human brain has been a focus of study for hundreds of years. Man has had a long standing fascination with the workings of the brain, usually with the motivation of exploring the possibility of creating an 'artificial being' with similar capabilities to humans in terms of thinking and understanding.

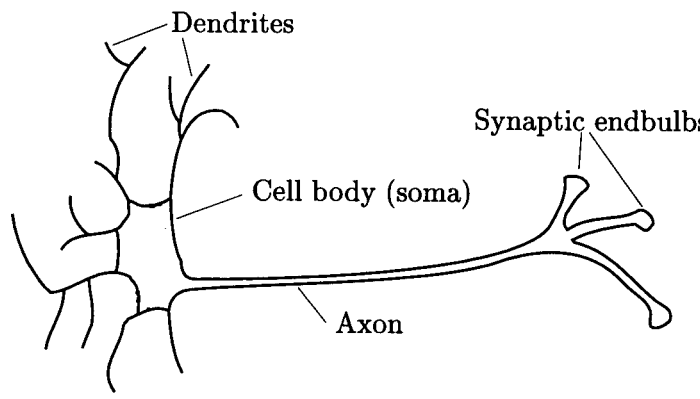


Figure 3.1 A biological neuron

Research has shown that the brain consists of a large number, typically 10^{11} of special cells called *neurons*. There are many different kinds of the neuron in a biological system, but certain features are common to them all; as represented in Figure 3.1. The cell body, the *nucleus* or *soma*, has one main connector, the fibrous *axon*, which acts as an output carrying signals to other neurons. This axon branches into many *synaptic endbulbs* at the ends of these fibres. Each neuron will have many *dendrites*, fibres leading to the neuron which receive signals from the axons of other neurons. There may be around 10^4 of these connected to any given neuron; the connections, known as *synapses*, being the meeting points of dendrites and synaptic endbulbs.

Although there is still not a full understanding of the workings of the brain, that which has been established has been part of the inspiration behind the various artificial neural network models presented here.

3.2.2 Threshold Logic Unit

The Threshold Logic Unit (TLU) was devised by McCulloch and Pitts[48] as one of the first neural processing elements. Graphically it may be represented as shown in Figure 3.2

The TLU has n input signals x_1, x_2, $x_3 \ldots x_n$ each with an associated weight w_1, w_2, $w_3 \ldots w_n$. It is a binary valued device on both the inputs and output with the output value being generated when a weighted sum of the inputs called the *activation*,

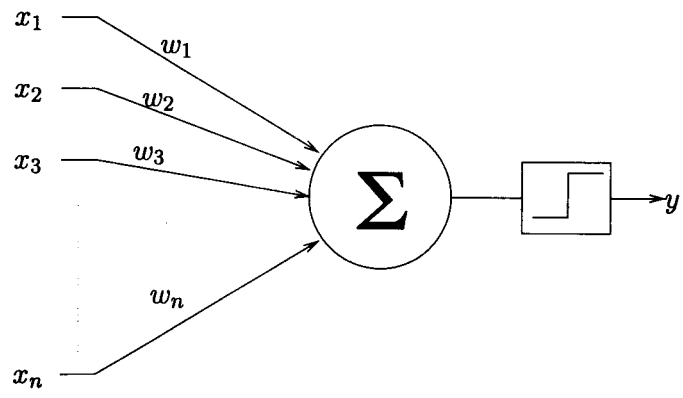


Figure 3.2 The Threshold Logic Unit

a, is passed through a threshold function according to

$$a = \sum_{i=1}^{n} w_i x_i \ , \tag{3.1}$$

and

$$y = \begin{cases} 1 & \text{if } a \geq \theta \\ 0 & \text{otherwise} \end{cases} \ . \tag{3.2}$$

The threshold, θ, is often zero and the threshold function acts as a *hard limiter* to produce the required binary output. That is, an activation value greater than or equal to the threshold gives a 1 output and an activation value less than the threshold will give a 0 output.

If the values on the inputs represent some pattern that is to be classified into one of a number of classes then the possible range of values on the inputs may be referred to as the *pattern space.* Each TLU with n inputs forms an n-dimensional decision-hyperplane[1] separating the pattern space into two regions or classes. This makes the TLU a linear classifier where any given input pattern is in either one of two classes depending on which side of the hyperplane it lies. By the use of more than one TLU in a network, multiple hyperplanes are defined giving rise to the ability to separate the inputs into more than two classes. Section 3.4.2 describes the process used to train a

[1] A hyperplane is a plane in n-dimensional space.

TLU.

3.2.3 The Perceptron

The perceptron is an enhancement of the TLU introduced by Rosenblatt[66] which was primarily designed as a pattern recognition system. Essentially it is a TLU whose inputs come from a set of association units. These association units can be assigned any arbitrary Boolean functionality but are fixed - they do not learn. The association units combine a group of the inputs from the pattern to be recognised and are designed to respond to certain features within the input pattern. Training is performed using the method described in section 3.4.2 which is known as the Perceptron learning algorithm as Rosenblatt was the first to use this training method.

3.2.4 Sigmoidal functions

The hard limiting threshold function on the output of the TLU is often replaced with a differentiable non-linear function of the activation. This is usually a curve from the family of so called *sigmoidal* curves such as the unipolar sigmoid, or logistic function

$$y = f(a) \triangleq \frac{1}{1 + e^{-a\rho}} , \tag{3.3}$$

to give a real valued artificial neuron.

The shape of this curve is determined by ρ with a larger value giving a sharper curve. Frequently ρ is set to 1 as is done in appendix A for calculating the derivative of f. A collection of typical unipolar sigmoid function responses are shown in Figure 3.3. An alternative to the unipolar sigmoid is the bipolar sigmoid function, or hyperbolic tangent function

$$y = f(a) \triangleq \frac{2}{1 + e^{-a\rho}} - 1 , \tag{3.4}$$

examples of which are shown in Figure 3.4. The use of unipolar or bipolar outputs has implications for the training of the neuron in that the bounds of the activation function should be matched to the bounds of the target values for neuron output.

With functions such as these on the output, the neuron is now able to work with

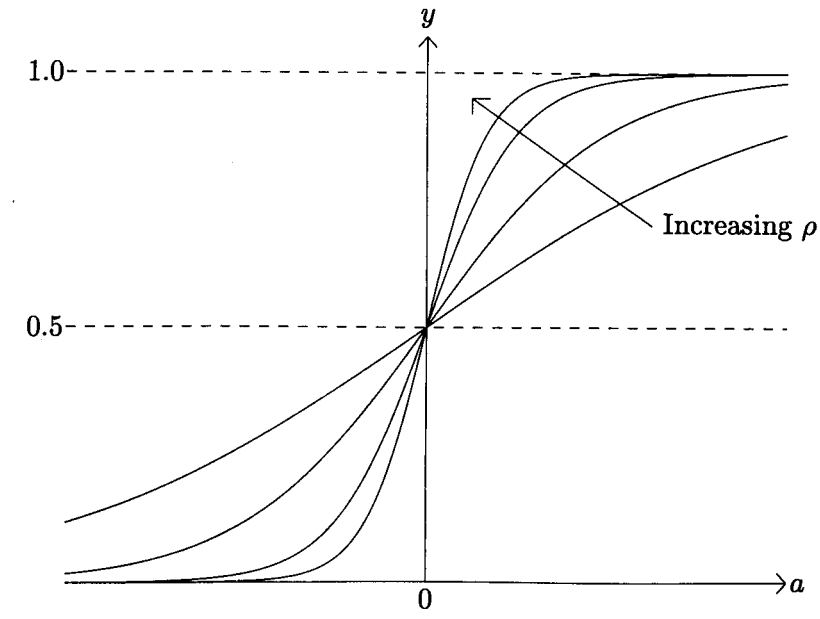


Figure 3.3 Typical unipolar sigmoid functions

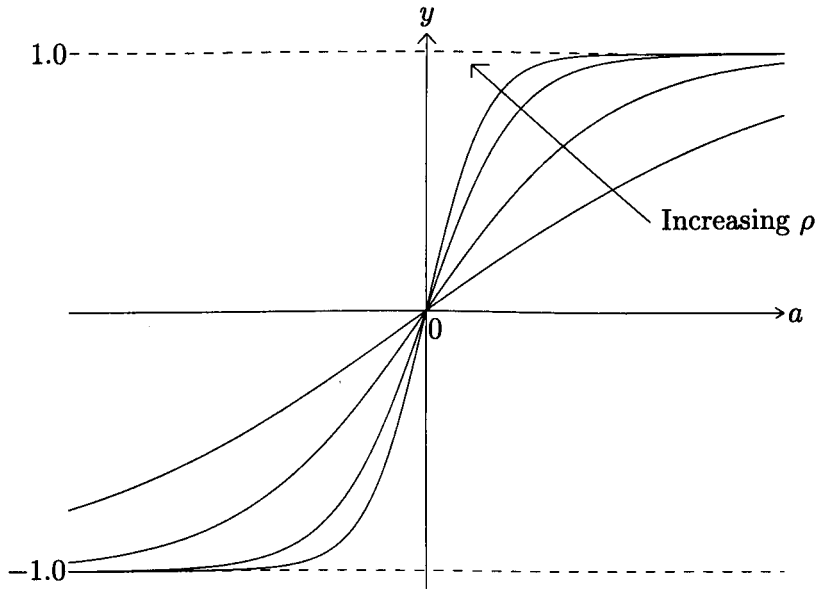


Figure 3.4 Typical bipolar sigmoid functions

more than just simple binary values; the neuron may now accept real values on the inputs and will give a real valued output.

3.3 Network Architectures

Thus far, consideration has been given to single neurons. As has been stated, these provide a single decision hyperplane within the input space and hence act as linear classifiers. To produce a system which may be applied to practical problems which are typically non-linear, neurons must be connected together to form more complex systems or *networks*. Just as there are different neuron models, so there are also different methods of interconnecting neurons to form networks.

The different network architectures have different processing capabilities, for example, early work by Minsky and Papert[53] showed that networks consisting of a layer of input neurons distributing data to a layer of output neurons could not solve certain simple problems such as the exclusive-OR and parity. Lippmann produced a review of common neural network techniques including discussion of their different capabilities in [46]. Consideration will now be given to some of the architectures in current use.

Artificial neural networks may be categorised in a number of different ways. One division is between the networks which work with binary signals and those which can take continuous valued inputs. Examples of the former are Hopfield[32] networks which can be used as an associative memory or a classifier, and Hamming networks which are optimum classifiers making use of the Hamming distance[21] in its classification.

In facial feature classification, the input signals from face images are continuous rather than binary, therefore attention will be given to networks that accept such signals. Within the binary / continuous valued divisions, ANNs may be said to fall into one of two basic types; supervised or unsupervised. Supervised networks are designed and trained in a manner such that the outputs are adjusted to give certain predetermined values depending upon the input values. Unsupervised networks have no predetermined connection between input and output values. Rather the network is designed to learn groupings within the data sets and use those in determining the output values. The architectures considered in this section will all be of the supervised

type. Unsupervised networks are examined in Section 3.6.

There are many kinds of continuous valued neural network in existance. To evaluate all the possible architectures with the problems of facial feature classification would be beyond the time scale available for this work. Therefore a selection of the possible architectures has been chosen according to their known strengths. The ones that have been used in this thesis have been selected because they are well known as classifiers, which is the fundemental funtion to be performed in this work. Other network architectures are better suited to different applications such as plant control, time series prediction where the learnt function within the neural network may be adapted over time.

3.3.1 Multilayer Perceptron

Simple two-layer networks where a set of input neurons distribute data via weighted connections to a set of output neurons as shown in Figure 3.5 perform a good job of mapping groups of similar input patterns onto similar output patterns. However, they do not work well when the inputs and outputs are very different so as to require a separate internal representation. This was overcome with the development of the multi-layer perceptron (MLP). With this arrangement, an additional layer of neurons is added between the input and output layer, known as the hidden layer since neither its inputs or outputs are visible to the outside world. It has been shown that a three layer network, such as the one shown in Figure 3.6, with an input layer, a hidden layer and an output layer is sufficient to approximate any function to a given degree of accuracy[46].

The problem that arises with the MLP is that of how to train the network as the desired values for the outputs of the hidden nodes are not known; so how can the error be measured if there is no knowledge of what the correct value should be? It was this problem that virtually halted work on artificial neural networks for several years. Eventually it led to the development of the backpropagation of error algorithm[87, 68] for modifying the weight values as given in 3.4.4.

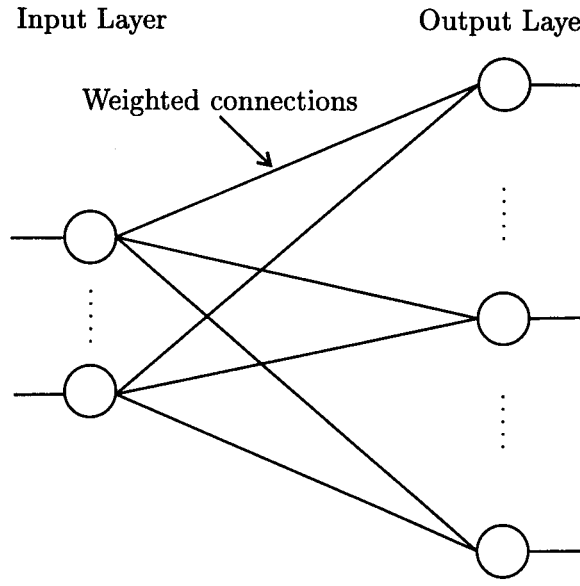


Figure 3.5 A two layer network

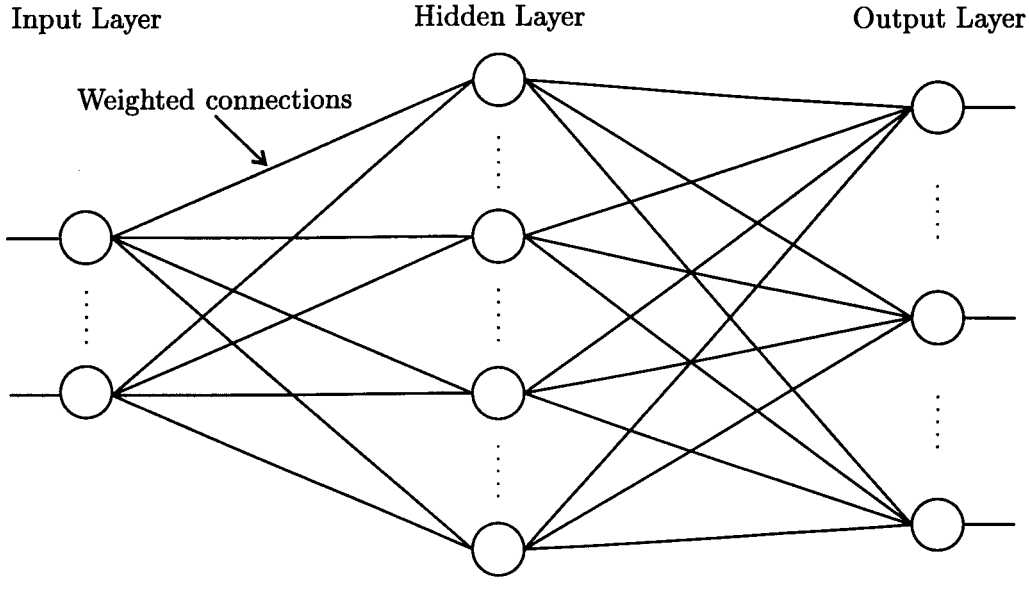


Figure 3.6 A three layer network

3.3.2 Radial-Basis Function Networks

A Radial Basis Function Network[57] (RBFN) introduces another type of neuron for use in its hidden layer. These neurons that provide the internal representation of the input vector are radially symmetric. For a function to be radially symmetric, it must have a number of qualities, namely:

- A centre, which, in the case of an RBF neuron, is a vector in input space.
- A distance measure, typically the Euclidean distance, giving the distance from an input vector to the centre.
- A transfer function which gives the output as a function of the distance measure. This is often a Gaussian function which gives a large output when the distance measure is close to zero and a small output as the magnitude of the distance measure increases.

Given the centre of the pattern, $\mathbf{c}_k$, for a given node k, and input vector, $\mathbf{x}$, the Euclidean distance

$$d(\mathbf{x}) = (\mathbf{x} - \mathbf{c}_k)^\top (\mathbf{x} - \mathbf{c}_k) \ , \tag{3.5}$$

between $\mathbf{x}$ and $\mathbf{c}_k$ can be calculated. This distance may be taken as being the activation of the hidden neuron k and this is then passed though a transfer function such as the Gaussian

$$y_k(\mathbf{x}) = \exp\left(\frac{d(\mathbf{x}) - r^2}{\sigma_k^2}\right) \tag{3.6}$$

to produce the neuron's output.

A typical RBFN is shown in Figure 3.7. The layer of radial basis function neurons is connected to the input layer and its output is then connected to a layer of regular perceptron units. These output neurons may have a linear transfer function and thus act as simply a weighted adder or they may use one of the non-liner transfer functions already discussed in Section 3.2.4. Training of a radial basis function network is discussed in Section 3.4.5

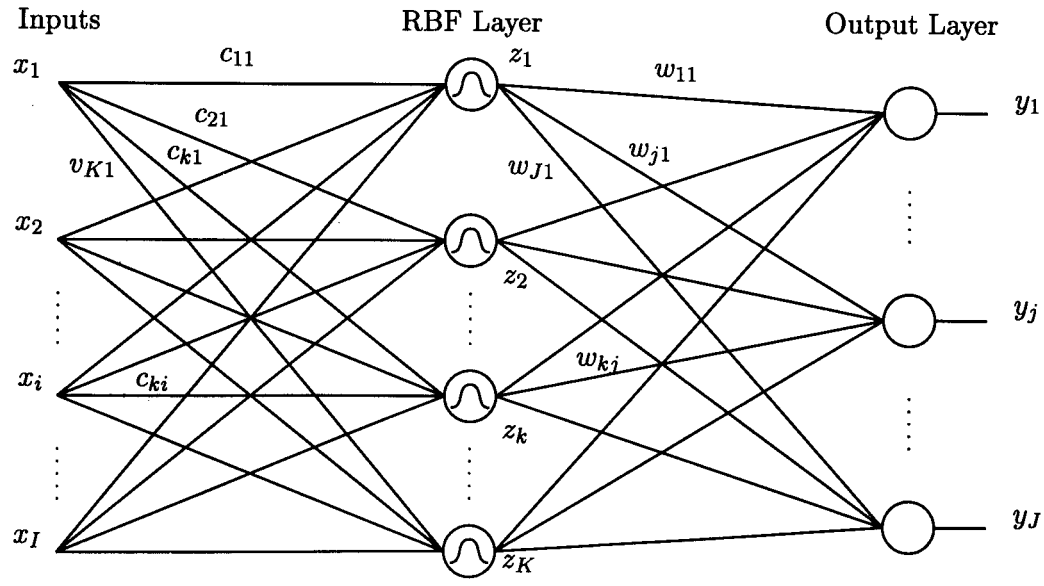


Figure 3.7 A radial basis function network

3.4 Training Algorithms

Thus far, the concept of training has not been considered in detail; let us now address this issue. Any neural network system is operated in one of two modes. In the first, the *training* mode, the network is presented with a set of data which is used to modify its interconnecting weights. This is the first phase of using any neural system. Usually, it is performed only once and then the network is used in the second, *recall* mode where the weight values are frozen and data is simply propagated through the network to give the output values. This is the deployment / use state of ANNs. In some systems, the training continues interactively with recall to enable the system to adapt to changing conditions. This form of system will not be considered here.

Having established some of the various forms of neural network model, methods need to be examined by which they may be trained and thus put to use. It is the case that there is no fixed method for training any particular network architecture, rather that there are a series of architectures and various learning methods have been developed to use with each one. Other methods are possible and some of the methods are applicable to more than one architecture.

3.4.1 The General Learning Rule

The process of training a neural network is that of altering the weights within the network according to some given rule. There is a general rule for this process which is the basis of most neural network training algorithms.

> The weight vector $\mathbf{w}_i \triangleq [w_{i1}\ w_{i2}\ \ldots w_{in}]^\top$ is changed in proportion, α, to the product of the input $\mathbf{x}$ and learning signal r.

The learning signal r is a function of the input $\mathbf{x}$, the current weight vector $\mathbf{w}_i$ and often the desired output from the node t_i. This results in the change in the weights at time τ being given using

$$\Delta \mathbf{w}_i^\tau = \alpha r\left[\mathbf{w}_i^\tau, \mathbf{x}^\tau, t_i^\tau\right]\mathbf{x}^\tau \ , \tag{3.7}$$

thus the altered weight vector becomes

$$\mathbf{w}_i^{\tau+1} = \mathbf{w}_i^\tau + \alpha r\left[\mathbf{w}_i^\tau, \mathbf{x}^\tau, t_i^\tau\right]\mathbf{x}^\tau \ . \tag{3.8}$$

3.4.2 The Perceptron Learning Algorithm

During the training of a TLU, the actual output, y, is compared to the desired output, t. When a misclassification occurs, $\mathbf{w}^\tau$ is altered according to the general learning rule (section 3.4.1), where the learning signal is taken as the difference between the desired output and the actual neuron's output. This gives

$$r \triangleq t_i - y_i \ , \tag{3.9}$$

with the threshold function on the output giving $y_i = \mathrm{sgn}(\mathbf{w}_i^\tau \mathbf{x}^\tau)$. Hence the weight adjustments are achieved by

$$\mathbf{w}_i^{\tau+1} = \mathbf{w}_i^\tau + \alpha\left[t_i^\tau - \mathrm{sgn}(\mathbf{w}_i^\tau \mathbf{x}^\tau)\right]\mathbf{x}^\tau \ . \tag{3.10}$$

This rule is only for binary valued neurons, in particular, (3.10) is for the bipolar case.

The weight adjustment may then be used in Algorithm 3.1 known as the *Perceptron learning algorithm*:

Algorithm 3.1 Perceptron learning algorithm

repeat
  for each training vector pair ($\mathbf{x}$,t) **do**
    evaluate the output y when $\mathbf{x}$ is input to the TLU
    if $y \neq t$ **then**
      form a new weight vector $\mathbf{w}^{\tau+1}$ according to (3.10)
    else
      do nothing
    end if
  end for
until $y = t$ for all vectors

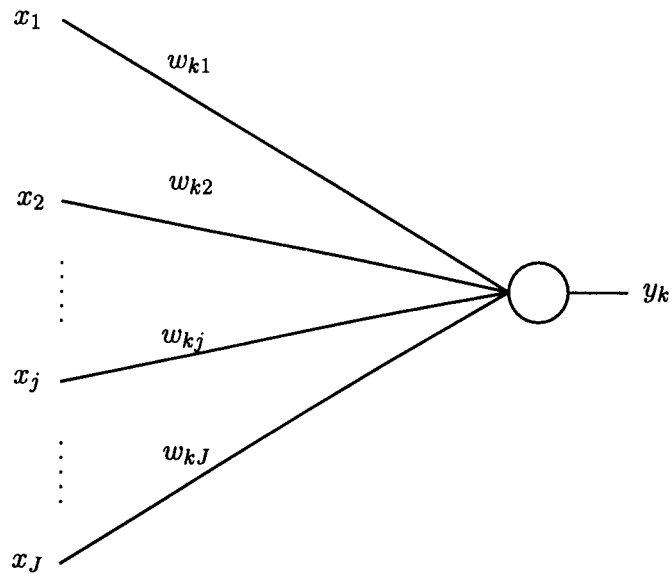


Figure 3.8 A single perceptron

3.4.3 Delta rule

In this discussion, consideration will be given to the perceptron shown in Figure 3.8. It may be said that the difference between the desired output from a network and its actual output for a given input is the error produced by that network. The error of the network, E, is a function of the weights since the outputs are functions of the weights. For a single node and input pattern p, this is

$$E_p = \frac{1}{2}(t_k - y_k)^2 \ . \tag{3.11}$$

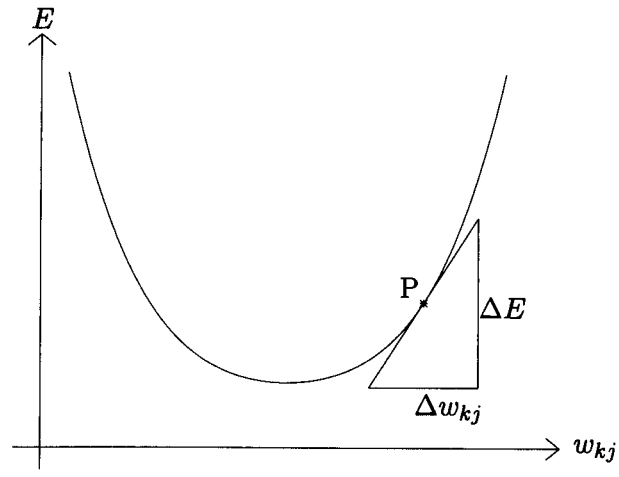


Figure 3.9 Gradient as a tangent to the curve

E is then the sum of these errors for all patterns

$$E = \sum_p E_p \ . \tag{3.12}$$

E is taken as the cost function that is to be minimised by altering the weights during the training process. Each weight needs to be altered in such a way that its contribution to the error is decreased. This may be done using the *gradient descent* method applied to the function relating the error to the weights.

Gradient Descent

In the gradient descent method, the gradient of the function at the current weight value is examined and used to determine a change in weight value that will reduce the error.

Given that $E = f(w_{kj})$, the gradient of $f()$ is the gradient of the tangent to the function at that point as shown in Figure 3.9 so

$$\text{gradient of } f() \text{ at P} = \frac{\Delta E}{\Delta w_{kj}} \ . \tag{3.13}$$

If Δw_{kj} is sufficiently small then $\Delta E \approx \delta E$, where δE is the change resulting in E

when w_{kj} is altered by Δw_{kj}, leading to

$$\delta E \approx \Delta E = \frac{\Delta E}{\Delta w_{kj}} \Delta w_{kj} \ , \tag{3.14}$$

giving

$$\delta E \approx \frac{\partial E}{\partial w_{kj}} \Delta w_{kj} \ . \tag{3.15}$$

Setting

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} \ , \tag{3.16}$$

where $\alpha > 0$ but is small enough to ensure that $\delta E \approx \Delta E$, gives

$$\delta E \approx -\alpha \left(\frac{\partial E}{\partial w_{kj}} \right)^2 \ , \tag{3.17}$$

which will give $\delta E < 0$ so by altering w_{kj} by Δw_{kj} we move down the curve towards the minimum.

Gradient descent for training neural networks

Following on from this, if gradient descent is to be used in training a neuron, then the error must then be a continuous, differentiable function of the weights such as the sigmoidal functions presented in section 3.2.4. Analysis of the differentiation of these functions is given in appendix A.

The error from a single node as given in (3.11) now becomes

$$E_p = \frac{1}{2} \left[t_k - f\left(\mathbf{w}_k^T \mathbf{x} \right) \right]^2 \tag{3.18}$$

from which the gradient is obtained as

$$\nabla E = -(t_k - y_k) f'(\mathbf{w}_k^T \mathbf{x}) \mathbf{x} \ . \tag{3.19}$$

Using (3.16) to get descent on the error curve, the change in weights is set to be

$$\Delta \mathbf{w}_k = -\alpha \nabla E \tag{3.20}$$

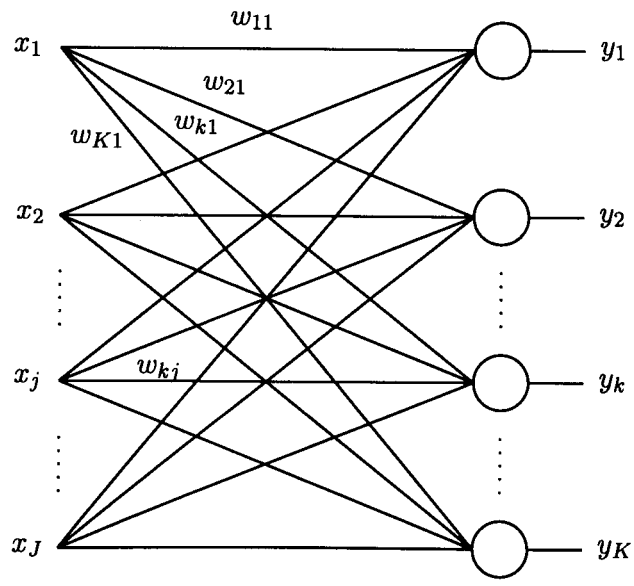


Figure 3.10 A single layer perceptron classifier

which using (3.19) gives

$$\Delta \mathbf{w}_k = \alpha(t_k - y_k) f'(\mathbf{w}_k^T \mathbf{x}) \mathbf{x} \tag{3.21}$$

so the adjustment of a single weight becomes

$$\Delta w_{kj} = \alpha(t_k - y_k) f'(\mathbf{w}_k^T \mathbf{x}) x_j \ . \tag{3.22}$$

Refering back to the general learning rule, the learning signal can now be defined as

$$r \triangleq [t_k - f(\mathbf{w}_k^T \mathbf{x})] f'(\mathbf{w}_k^T \mathbf{x}) \ . \tag{3.23}$$

Now this can be extended for the case of more than one neuron as shown in Figure 3.10. Given that the target output is now a vector $\mathbf{t} \triangleq [t_1 \ t_2 \ \ldots \ t_K]^\top$, the network error for a given pattern p is now

$$E_p = \frac{1}{2} \sum_{k=1}^{K} (t_k - y_k)^2 = \frac{1}{2} \|\mathbf{t} - \mathbf{y}\|^2 \ . \tag{3.24}$$

Following (3.20) the individual weight changes are calculated as

$$\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} , \tag{3.25}$$

where E is the error given in (3.24), omitting the p subscript. The activation a_k for each node in layer k is defined as

$$a_k = \sum_{j=1}^{J} w_{kj} x_j \tag{3.26}$$

giving the output of the neuron to be

$$y_k = f(a_k) \tag{3.27}$$

The *error signal term* δ from the k'th neuron is now defined as

$$\delta_{yk} \triangleq -\frac{\partial E}{\partial a_k} . \tag{3.28}$$

The term $\partial E / \partial w_{kj}$ is only dependent on the value of a_k for a single neuron since the error for the k'th output is produced only by the output y_k and hence the weights w_{kj}, $j = 1, 2, \ldots, J$ for a given value of k. So, by the chain rule

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial a_k} \cdot \frac{\partial a_k}{\partial w_{kj}} \tag{3.29}$$

The second term here is the derivative of (3.26), which since x_j, $j = 1, 2, \ldots, J$ are constant for a given input pattern, gives

$$\frac{\partial a_k}{\partial w_{kj}} = x_j . \tag{3.30}$$

So using (3.28) and (3.30), (3.29) can be written as

$$\frac{\partial E}{\partial w_{kj}} = -\delta_{yk} x_j . \tag{3.31}$$

The change in weights can now be calculated using

$$\Delta w_{kj} = \alpha \delta_{yk} x_j \ . \tag{3.32}$$

This is a general equation for the weight adjustment in the delta rule which is applicable regardless of the form of the activation function. It is the equation used in the calculation of the delta value, δ_{yk}, which varies according to the chosen activation function.

Using the chain rule on (3.28) gives

$$\delta_{yk} = -\frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial a_k} \ , \tag{3.33}$$

where the second term can be seen to be the derivative of the activation function

$$f'_k(a_k) \triangleq \frac{\partial y_k}{\partial a_k} \tag{3.34}$$

and also

$$\frac{\partial E}{\partial y_k} = -(t_k - y_k) \ . \tag{3.35}$$

Hence (3.33) can be re-written as

$$\delta_{yk} = (t_k - y_k) f'(a_k) \tag{3.36}$$

so the weight change formula now becomes

$$\Delta w_{kj} = \alpha (t_k - y_k) f'_k(a_k) x_j \ . \tag{3.37}$$

This formula can be used for any form of differentiable activation function $f(a)$. Two of the commonly used activation functions have been discussed in section 3.2.4.

3.4.4 Backpropagation

As mentioned in 3.3.1 there is a difficulty in training networks with more than one layer of processing elements in that the desired outputs of hidden layers are not known

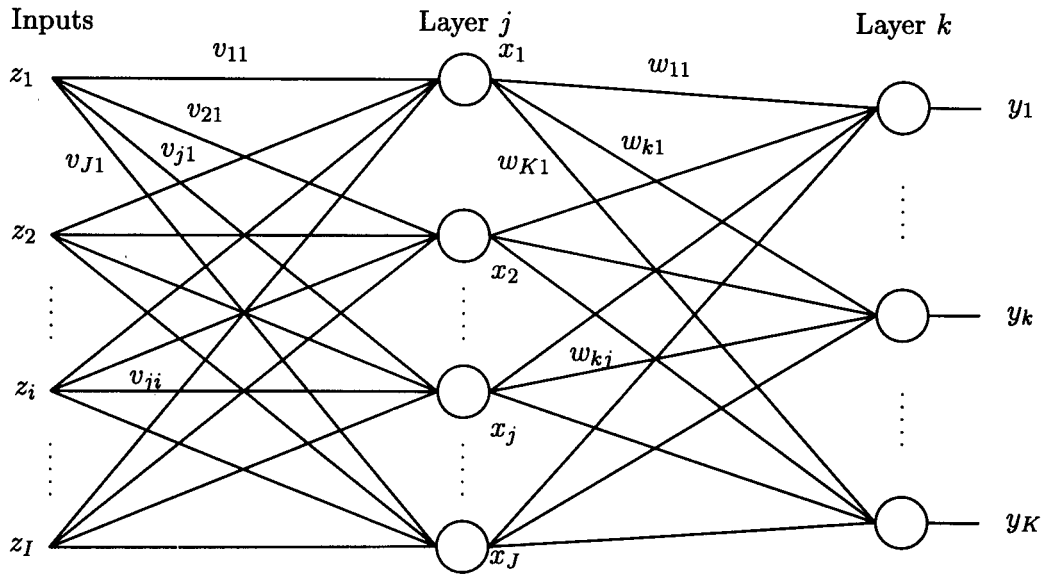


Figure 3.11 A multi-layer perceptron

so that method of measuring error is not possible. As a solution to this problem, the delta rule has been extended to form the generalised delta rule, also known as *backpropagation of error*[87, 68]. With this method, the error in an output from the network is propagated back through the network, adjusting the weights on the way in proportion to the amount that the weight in question contributed to the error.

Consider a network as shown in Figure 3.11. Layer j is known as the *hidden layer* as its outputs are not visible outside the network. For this layer, the gradient descent equation is

$$\Delta v_{ji} = -\alpha \frac{\partial E}{\partial v_{ji}} , \tag{3.38}$$

and (3.29) becomes

$$\frac{\partial E}{\partial v_{ji}} = \frac{\partial E}{\partial a_j} \cdot \frac{\partial a_j}{\partial v_{ji}} . \tag{3.39}$$

So the weight adjustment for the hidden layer may be written as

$$\Delta v_{ji} = \alpha \delta_{xj} z_i \tag{3.40}$$

with δ_{xj} being the error signal term produced by the j'th neuron of the hidden layer

with output $\mathbf{x}$. It may be defined as

$$\delta_{xj} \triangleq -\frac{\partial E}{\partial a_j} \ . \tag{3.41}$$

Applying the chain rule and proceeding as before

$$\delta_{xj} = -\frac{\partial E}{\partial x_j} \cdot \frac{\partial x_j}{\partial a_j} \tag{3.42}$$

with the second term being

$$\frac{\partial x_j}{\partial a_j} = f_j'(a_j) \ . \tag{3.43}$$

Unlike the nodes in the output layer, the error produced by a hidden layer node contributes to each component of the error sum in (3.24) so the first term of (3.42) is

$$\frac{\partial E}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{1}{2}\sum_{k=1}^{K}\left(t_k - f_k\left(a_k(\mathbf{x})\right)\right)^2\right) \tag{3.44}$$

which can be simplified to

$$\frac{\partial E}{\partial x_j} = -\sum_{k=1}^{K}(t_k - y_k)\frac{\partial}{\partial x_j}\{f(a_k(\mathbf{x}))\} \ . \tag{3.45}$$

Calculating the derivative results in

$$\frac{\partial E}{\partial x_j} = -\sum_{k=1}^{K}(t_k - y_k)f_k'(a_k)\frac{\partial f(a_k)}{\partial x_j} \tag{3.46}$$

which using (3.36) and (3.26) can be simplified to

$$\frac{\partial E}{\partial x_j} = -\sum_{k=1}^{K}\delta_{yk}w_{kj} \ . \tag{3.47}$$

Using (3.43) and (3.47), δ_{xj} from (3.42) can be rearranged to give

$$\delta_{xj} = f_j'(a_j)\sum_{k=1}^{K}\delta_{yk}w_{kj} \tag{3.48}$$

and so the weight change for the hidden layer now becomes

$$\Delta v_{ji} = \alpha f_j'(a_j) z_i \sum_{k=1}^{K} \delta_{yk} w_{kj} \ . \tag{3.49}$$

This now expresses the *generalised delta learning rule.* It can be seen that the change in weights leading to node j is proportional to the weighted sum of the δ values of all the nodes in layer k with the weights w_{kj} as the weighting factors. So for any given hidden layer node j, all the output layer errors, δ_{yk}, contribute to the adjustment of the weights v_{ji}.

The weight update expression for the hidden layer of the network in Figure 3.11 can be written more concisely in vector notation as

$$\mathbf{V}^{\tau+1} = \mathbf{V}^{\tau} + \alpha \boldsymbol{\delta}_y \mathbf{z}^{\tau} \tag{3.50}$$

where $\mathbf{V}$ is the matrix of weights v_{ji}, $\boldsymbol{\delta}_x$ is the column vector of error terms δ_{xj} and $\mathbf{z}$ is the column vector of inputs z_i. If the j'th column of the weight matrix $\mathbf{W}$ is defined as $\mathbf{w}_j$, $\boldsymbol{\delta}_x$ can be computed by

$$\boldsymbol{\delta}_x = \mathbf{w}_j^{\tau} \boldsymbol{\delta}_y \mathbf{f}_x' \tag{3.51}$$

3.4.5 Training an RBF network

For clarity, the figure of an RBF network given previously is repeated in Figure 3.12. The training of an RBF network such as this is a two stage process. First values must be assigned to the hidden layer basis function centres, $\mathbf{C} = \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$, and the widths of the basis functions, $\sigma_k, k = 1, \ldots K$. This process is then followed by training of the output layer of neurons. Training of the hidden layer neurons is usually performed by means of a clustering algorithm to determine the basis function centres and nearest neighbour heuristics are used to evaluate σ_k.

One method of performing clustering that is frequently used is the *K-means* clustering algorithm. This requires that all the training vectors are available to the algorithm. Let us consider a network as shown in Figure 3.12. Given the training set $\mathbf{X} \triangleq \mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_p$, we are to find the set, $\mathbf{C} \triangleq \mathbf{c}_1, \ldots \mathbf{c}_K$ of K cluster centres where

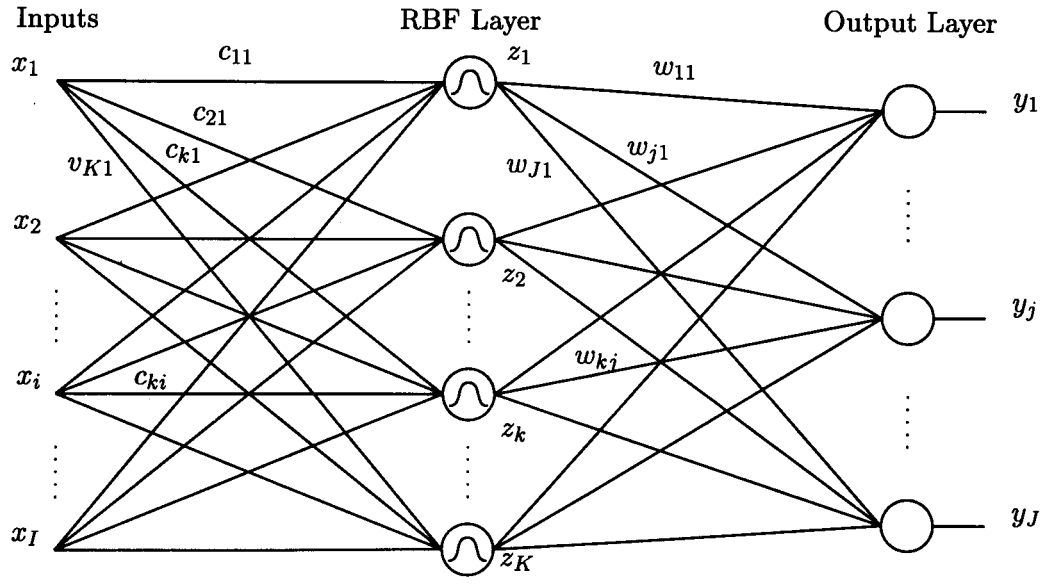


Figure 3.12 A radial basis function network

$\mathbf{c}_k \triangleq c_{k1}, c_{k2}, \ldots c_{kp}$. These centres are required to satisfy the condition that the sum of squares distance between each training pattern, $\mathbf{x}_p$ and its nearest cluster centre is a local minimum. This can be expressed by defining a cluster membership function $m(\mathbf{x}_p)$ as:

$$m(\mathbf{x}_p) = k \Leftrightarrow (\mathbf{c}_k \text{ is closest cluster centre to } \mathbf{x}_p) \ . \tag{3.52}$$

The *K-means algorithm* is initialised by assigning a random set of cluster centres to $\mathbf{C}$. Then the following steps are repeated for either a fixed number of iterations or until the membership function $m()$ remains constant.

- Create the membership function $m()$.
- Change the cluster centres by setting $\mathbf{c}_k$ to the mean value of all training vectors $\mathbf{x}_p$ which have a membership value of k i.e.

$$n\mathbf{c}_k = \sum_{\{p:m(\mathbf{x}_p)=k\}} \mathbf{x}_p \tag{3.53}$$

where n is the number of training vectors with membership value k.

Having found the cluster centres, the next step is to set the widths, σ, of the

transfer functions. This may be achieved by the use of P nearest neighbour heuristics. Considering the cluster centre $\mathbf{c}_k$, $k_1, \ldots k_P$ may be taken as the indices of the P nearest neighbouring cluster centres. Then the width of the transfer function is set as

$$\sigma_k = \sqrt{\frac{1}{P}\sum_{p=1}^{P} \| \mathbf{c}_k - \mathbf{c}_{kp} \|^2} \tag{3.54}$$

Using this technique, the hidden layer of a radial basis function network is trained purely by self-organisation since no reference is made to the desired outputs. Once the hidden layer is trained in this manner, the output layer can be trained using any of the algorithms applied to the training of a perceptron, typically the delta rule as discussed in Section 3.4.3.

3.5 Measuring the network's performance

Having established a means to train a neural network, a measure is needed to determine whether the network is doing anything useful or not.

3.5.1 Measures of performance

Generally there are two or three data sets of interest in work involving neural networks. There is the set that is used for training the network and there is usually also a set used to test the trained network. In addition there may be a third data set used to validate the network's performance. This validation data set is a second testing data set that is not involved in any way with the training process.

As training progresses, the weights are modified in such a way as to map the input vectors in the training data onto the corresponding output vectors. The error measure, taken as the difference between the actual and the desired outputs, that is used in the training of the network may be used as a measure of the networks performance. However if we only wanted a system to perform these mappings then a lookup table would be sufficient. In using a network we are usually wanting to generalise from the training data so that input vectors other than the ones used for training will produce the same outputs as similar vectors in the training data set.

A measure is needed of the network's performance on each of the two data sets, though usually more interest is paid to the results from the testing data. Two possible measures are *r.m.s. error* and *accuracy*.

r.m.s. error

The r.m.s. error method uses the errors in the value of the outputs for each of the given vectors applied to the network. These errors are totalled by means of the root-mean-square method where the errors are first squared to eliminate the effect of differing signs, then the mean is taken and the square root is taken to give the final answer. This measure is often used, as training progresses, as a measure of how well the network has converged to the required solution. As the errors in the outputs are used in the training process to update the weights, this is a simple calculation to perform.

Accuracy

Accuracy refers to the number of correct classifications that the network has achieved. This type of measure is only suitable where there are a discrete number of classes into which the data is to be partitioned. Under the condition where there are two classes, class *one* and class *two*, the following results may be defined:

True positive A correct identification of a pattern vector that belongs to class *one*.

True negative A correct identification of a pattern vector that belongs to class *two*.

False positive Identification of a pattern vector as belonging to class *one* when infact it belongs to class *two*.

False negative Identification of a pattern vector as belonging to class *two* when infact it belongs to class *one*.

Undetermined No conclusive output.

Then accuracy may be defined as

$$accuracy = \frac{true\ positives + true\ negatives}{all\ output\ cases} \times 100 \tag{3.55}$$

Alternatively, there is the case where there are several classes within the data. In this situation, the results fall into one of three classes

Correct A correct identification of a pattern vector that belongs to the class that the network has output.

False An identification of a pattern vector as belonging to one class when infact it belongs to another.

Undetermined No conclusive output.

Now, accuracy is defined as

$$accuracy = \frac{correct}{all\ output\ cases} \times 100 \tag{3.56}$$

The *undetermined* outputs arise from the use of a *threshold* on the output. Supposing that the network has one output where a value of 1.0 represents class *one* and a value of -1.0 represents class *two*. We need to define the meaning of outputs that lie between these two and this is where the threshold, T, is used

$$output\ class = \begin{cases} one & y > +T \\ two & y < -T \\ \text{undetermined} & \text{otherwise} \end{cases} \tag{3.57}$$

This threshold is needed since a system that simply took positive outputs as belonging to class *one* and negative as class *two*, would be sensitive to noise on the inputs as only small changes in the output value could cause the networks' classification to alter. Consideration must be given to the setting of this value in trading off noise immunity with the accuracy of the results achieved. Plotting accuracy against threshold will reveal something of how well a network is generalising; the more area under the curve, the better the networks' performance as it is producing output values closer to the target ± 1. An example of such a graph is given in Figure 3.13.

The accuracy measure may be calculated using any of the two, or three, data sets involved in the design of the ANNs to solve the problem. The best reflection of the

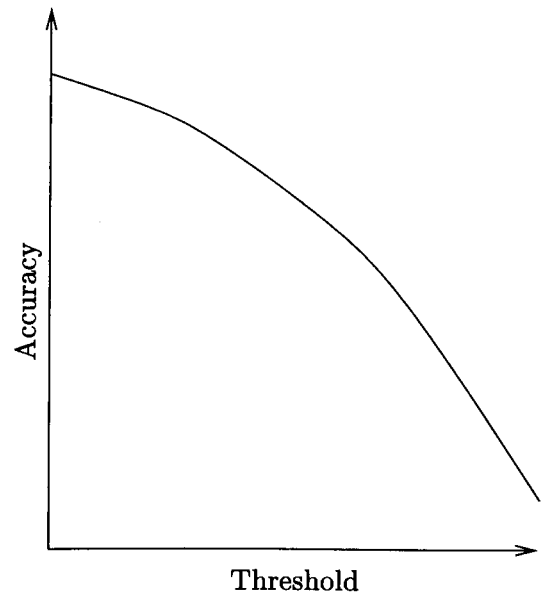


Figure 3.13 A typical plot of accuracy against threshold

performance of the network is the figure obtained by using the validation data set as this reflects the performance of the network when it is presented with previously unseen data. However if there is insufficient data available to make a validation data set then the accuracy achieved with the testing data set will give the best indication of the network's generalisation performance.

3.5.2 The Confusion Matrix

If all that is required, in terms of measuring the performance of a neural network, is a simple measure of how much it got correct, then the accuracy measure mentioned in 3.5.1 is sufficient. However, if the network is being trained to perform a classification with more than two classes, then a clearer representation of the network's performance is possible. The *confusion matrix* is such a device and shows for any given target class, what actual classifications are being made.

Table 3.1 shows an example confusion matrix. This one is for a four class problem, the classes being $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$ with the data set containing fifteen examples of each class. The first column of the matrix shows that of the 15 examples of $\mathcal{A}$, 12 were correctly identified as being in class $\mathcal{A}$, 2 were identified as belonging to class $\mathcal{B}$ and

Table 3.1 A typical confusion matrix

Actual		Target			
		$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$
	$\mathcal{D}$	1	0	4	**10**
	$\mathcal{C}$	0	0	**5**	1
	$\mathcal{B}$	2	**15**	5	3
	$\mathcal{A}$	**12**	0	1	1

one to class $\mathcal{D}$. The second column shows that all 15 examples of class $\mathcal{B}$ were correctly identified as belonging to class $\mathcal{B}$ etc. The correct classifications all appear on the leading diagonal and to aid clarity of reading, these figures will all be printed in bold typeface in confusion matrices presented in this thesis.

From such a matrix, the accuracy measure may be calculated as

$$accuracy = \frac{\sum leading\ diagonal}{\sum all\ elements} * 100\ . \tag{3.58}$$

In the case where a threshold is being used and thus there is the possibility of vectors being unclassified, this condition can be represented as an additional class within the confusion matrix. The difference here being that there will be no entries in the "target" column for the "undetermined" class and therefore this class need only be entered as a row in the matrix.

A confusion matrix enables observation to be made as to any classes that are particularly difficult for the network to distinguish. For example in the matrix given in Table 3.1 there class $\mathcal{C}$ has been poorly identified, with many of its examples being classified as either class $\mathcal{B}$ or class $\mathcal{D}$.

As with the accuracy measures, a confusion matrix may be applied to any of the three data sets involved in the problem. The confusion matrix is particularly relevant in multi class problems although it can also be helpful in two class problems where a threshold results in the third case of "undetermined" in determining which of the classes is more easily identified.

Other measures may be made from the outputs of artificial neural networks to gauge their performance such as sensitivity and specicivity. These are related particularly to networks where there are two just possible output values. They are used in cases where,

in the case of sensitivity for example, it is more important that all positive outputs are correct than it is for all negative outputs are correct. These measures, all of which are calculated from the number of true and false positive and negative results, are not of great relevance to the work in this thesis. In the instances where there are two just different output values, neither of the conditions is more important than the other, the overall accuracy is the most significant part.

3.6 Unsupervised Neural Networks

Thus far, the neural network models described have fallen into the category known as *supervised* networks. In these, the desired response for a given input pattern vector is known and the training process is one of finding a set of weights that will cause the network to perform this mapping. An alternative type of network is an *unsupervised* or *self organising* network where no desired outputs are given. In this case, the network assigns the classifications to the input patterns, usually grouping similar vectors together into *clusters*. One such form of network is the *Kohonen* network[40] or *Self-Organising Map*. The basic operation of this type of network is now described.

3.6.1 Kohonen network

Looking at it in the most simplest sense, the Kohonen network[40] or Self-Organising Map (SOM) performs the task of clustering data into groups containing similar pattern vectors. A SOM network usually consists of a two dimensional layer of neurons which maps the input n-dimensional input space onto an ordered, 2-dimensional map. This layer is connected to an input layer which is purely for distributing the data into the network. A typical SOM network is shown in Figure 3.14; note that the input layer is fully connected to the Kohonen layer though only some of the connections are shown in this diagram.

The neurons in the SOM layer are similar to those in the hidden layer of a radial basis function network (see Section 3.3.2) in that they calculate the Euclidean distance from their weights to the input pattern vector. In the normal case, the neuron with the smallest distance value, known as the *winning* neuron, will give an output of 1.0

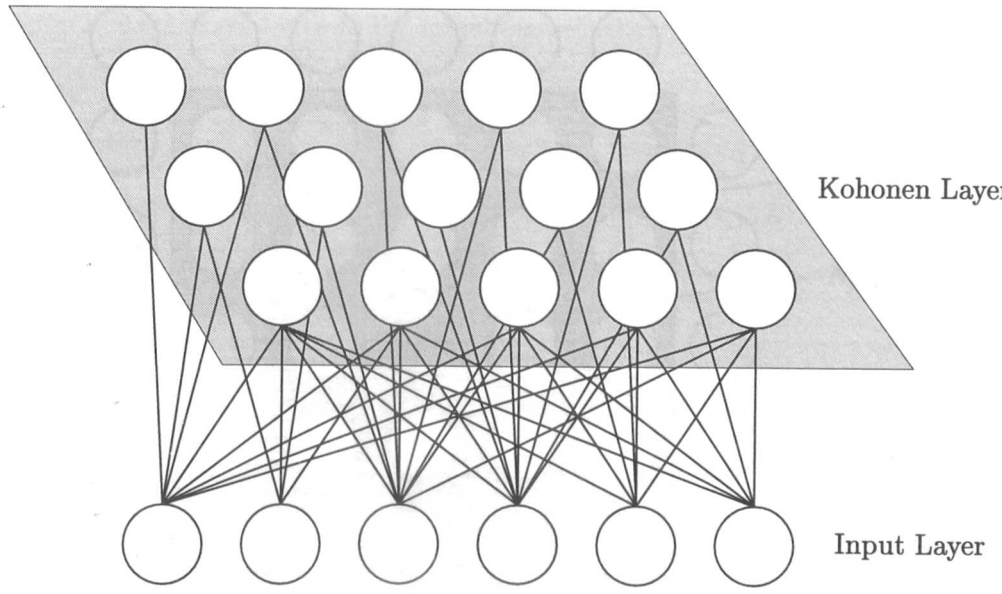


Figure 3.14 A Kohonen Self-Organising Map

while all the others will produce 0.0, or alternatively, schemes have been devised where a number of neurons with small distance values will generate positive output values by the use of some form of interpolation.

Given the J valued input pattern vector $\mathbf{x} \triangleq [x_1, x_2, \ldots x_J]$ then each neuron, k, in the Kohonen layer will have J weight vectors, collectively denoted by $\mathbf{w}_k \triangleq [w_{k1}, w_{k2}, \ldots w_{kJ}]$. The Euclidean distance, D_k, is calculated for each of the K Kohonen neurons according to

$$\| \mathbf{x} - \mathbf{w}_k \| = \sqrt{\sum_j (x_j - w_{kj})^2} \tag{3.59}$$

and the winning neuron, m, is found such that

$$\| \mathbf{x} - \mathbf{w}_m \| = \min_k (\| \mathbf{x} - \mathbf{w}_k \|) \ . \tag{3.60}$$

During the training of a SOM[40], the weight values of the winning neuron are adjusted in order to further reduce its distance from the input pattern vector. However, it is not only the winning neuron which undergoes this adjustment. A neighbourhood of neurons surrounding the winning one are also adjusted. This neighbourhood, $N_m(\tau)$, is defined as the neurons within a radius, r, of the winning neuron and this radius is decreased

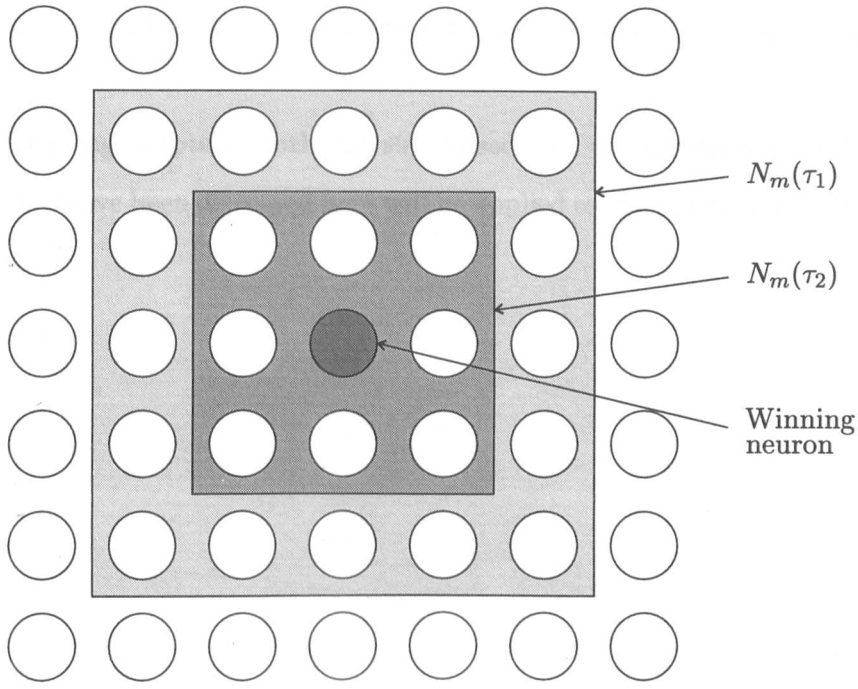


Figure 3.15 Neighbourhood regions in a Kohonen layer

as the training progresses. An example of neighbourhood is shown in Figure 3.15. At time τ_1 the neighbourhood includes all neurons within the light grey box whereas at time τ_2, it is only those within the darker grey box that are considered to be within the neighbourhood. It is this method of adjusting weights of neurons surrounding the winning neuron that gives the SOM its particular property of ordering the pattern vectors such that the winning neuron for a given input vector will be in the same area of the map as the winning neuron for similar input vectors. As with other networks, a learning rate, α is used to govern the change in the weights. However here we require it to decrease as learning progresses so it is often dependent on the neighbourhood radius, r, and the training time, τ. The weight update rule for Kohonen neurons then is

$$\Delta \mathbf{w}_k(\tau) = \alpha(N_k, \tau)\,[\mathbf{x}(\tau) - \mathbf{w}_k(\tau)] \text{ for } k \in N_m(\tau)\ . \tag{3.61}$$

A SOM may be used in its own right as a means of clustering a data set or, it could have a supervised network layer attached to its output, which can be trained once the unsupervised clustering has been completed. This approach can be used to

match clusters within the data to the target classifications associated with each pattern vector.

In the following chapters, both the supervised and unsupervised neural network techniques that have been discussed here will be applied to the problem of facial feature classification.

Chapter 4

Experimental Data, Pre-processing and Analysis

4.1 The Data

The data used for this thesis consisted of a set of 1000 face images of Caucasian males aged 18-65 and was originally collected by the Department of Psychology at Aberdeen University[73]. Associated with each face image, was a set of 50 descriptive measures and 37 feature points marking certain positions on the facial image.

4.1.1 The descriptive measures

We do not currently know of a unique way to describe a face and the approach selected is largely governed by the use of prior knowledge in order to implement a realistic solution. There is a great deal of subjectivity involved as what one person may describe as a long face, another may call normal size. However, in order to use a description as a key in to an index of images, a fixed set of measures is needed. Such a set of measures has been designed by the Department of Psychology at Aberdeen University[73] based on previous work by the same group[76]. There are 50 of these measures given in Table 4.1(physically derived measures) and Table 4.2(non-physically derived measures), some of which are in the form of "yes/no" answers and some are ratings on a scale of 1–5.

Table 4.1 Physically derived measures

Feature	Property	Scale
Face	length	1–5
	width	1–5
Hair	length	1–5
	thickness	1–5
Forehead	height	1–5
	width	1–5
Eyebrow	thickness	1–5
	setting	1–5
	height	1–5
Eyes	size	1–5
	narrow/open	1–5
	setting	1–5
Nose	length	1–5
	width	1–5
	tip	1–5
Mouth	size	1–5
Upper lip	thickness	1–5
Lower lip	thickness	1–5
Chin	size	1–5
	shape	1–5

4.1.2 Data acquisition

A photographic studio was set up by the researchers at Aberdeen to take the photographs under controlled conditions, and volunteers for the photographs were approached both at the university and in the town. Each volunteer was dressed in a surgical gown to prevent any differences occurring in the images as a result of variations in clothing, and a height adjustable chair with a neck support was used to ensure that the photographs all had the faces located in the same position[73]. The original photographs were recently digitised by a research group at the Home Office to produce 24 bit colour images at a resolution of 384×512 pixels.

Having obtained the images, the Aberdeen group obtained the 50 descriptive measures described in Section 4.1.1 for each face by presenting the images to a group of jurors who had been trained to recognise the categorisations that were required.

In addition the positions of a set of 37 feature points were measured on the photographs by the use of a graphics tablet and the data was then stored on a computer.

Table 4.2 Non physically derived measures

Feature	Property	Scale
Shape of face	bony – fleshy	1–5
Complexion	fair – dark	1–5
	pale – florid	1–5
	unlined – lined	1–5
	clear – blemished	1–5
Hair	tidy – untidy	1–5
	straight – curly	1–5
	no grey – white	1–5
	black – blond	1–5
Forehead	straight – sloping	1–5
Eyebrows	straight – bent	1–5
Eyes	deep set – protruding	1–5
	blue – brown	1–5
	small – large	1–5
Nose	small – large	1–5
	concave – hooked	1–5
	narrow tip – broad tip	1–5
Chin	receding – jutting	1–5
Facial hair	none at all	yes/no
	moustache	yes/no
	sideburns	yes/no
	beard	yes/no
Squint		yes/no
Bags under eyes		yes/no
Scars		yes/no
Glasses		yes/no
Earring		yes/no
Age		
Weight		
Height		

The points that were chosen are shown in Figure 4.1. For the research in this thesis, the descriptive measures were available from the orginal Aberdeen research but the locations of the feature points were not. The feature points that were relevant to the research were re-digitised by the author as described in Section 4.2.2.

Measures such as distances, angles and areas taken from these points were correlated with the human estimated descriptors and the original descriptors that showed high correlation with the new measures were then replaced by these mathematically derived ones. This list of descriptive measures is given in Table 4.1. For work on automatically extracting some of the "physical" features using conventional image processing techniques, see [13].

While some of the measures can be mathematically derived from the positions of feature points, the work presented in this thesis is concerned with investigating the measurement of features that do not fall into this category. The complete list of these measures is given in Table 4.2. The aim here is to use artificial neural network techniques to 'derive' these measures.

4.2 Pre-processing

The raw data for the 1000 faces was supplied in 24-bit colour as a series of TIFF[1] images each 590,028 bytes in size. The classification of facial features, is usually concerned with only a small part of the data contained within such images. Therefore it would be inappropriate to present the whole of an image to an artificial neural network or any other data analysis technique as the information relating to the feature in question may well be masked by the surplus information supplied. In other words, with such a large quantity of information being supplied, the network or analysis technique may not isolate the data relevant to the current classification. So methods were needed to extract suitable data from the images and manipulate it into a form appropriate for subsequent analysis.

Since the original data was supplied in colour, one of the major issues here is that

[1]TIFF stands for Tagged Image File Format. TIFF is a very flexible format for storing bitmapped images in up to 24-bit colour. Information for decoding these files was taken from[65].

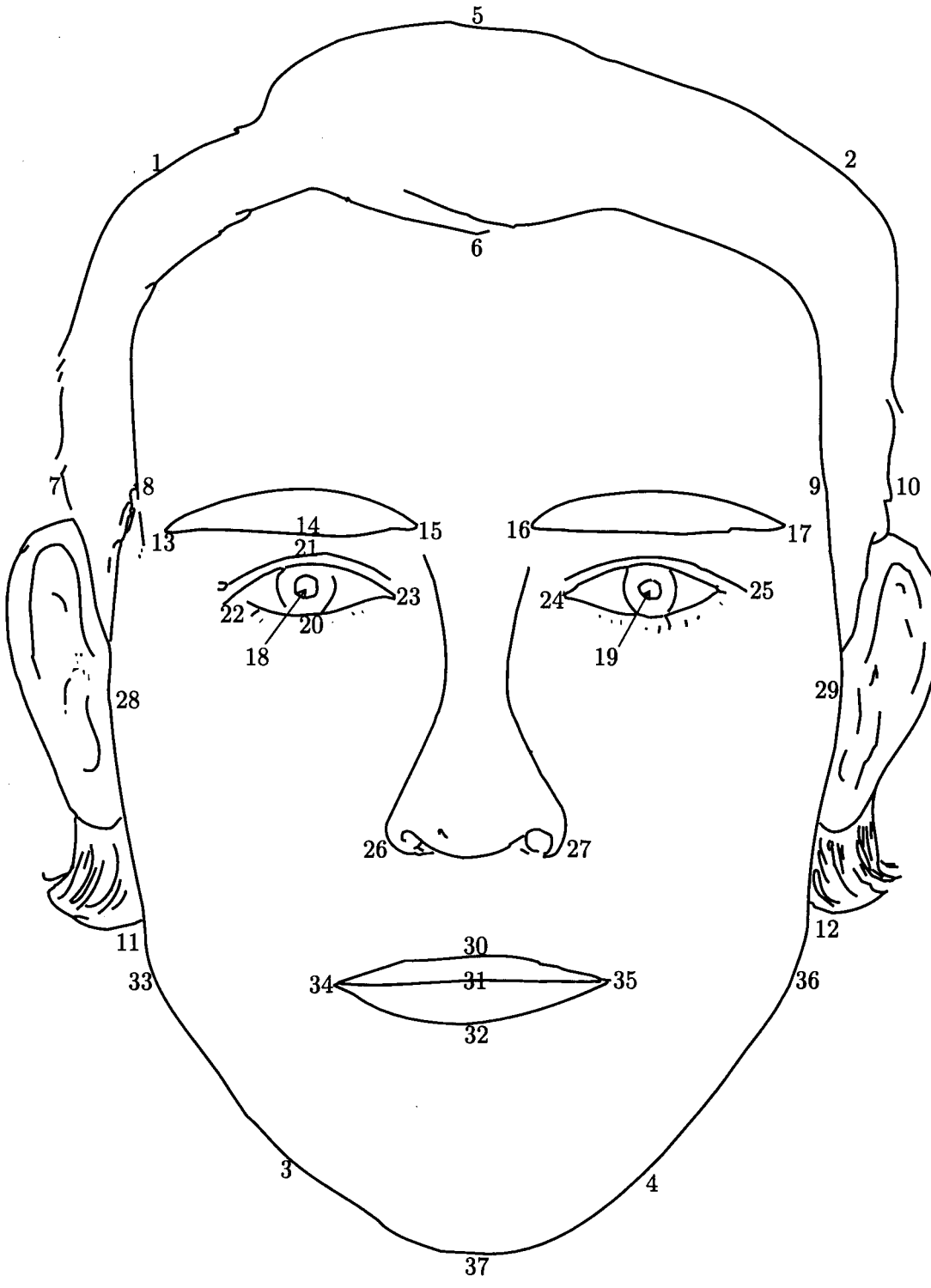


Figure 4.1 Points recorded on face images

of colour representation within the digital image as there are many alternative methods available. Some representations will highlight given features better than others, whereas a different representation may be needed if the feature to be classified is the colour of an image section rather than, say, a shape. The other main issue is that of how much of the image to present to the network. Naturally if we are only concerned with the eye, there is little point in presenting the whole image to the network; in fact this is likely to reduce the network's ability to correctly identify the classification of the image feature as there are no markers pointing to the area in question. So the issue arises as to how to segment the image for presentation to the network and this will be addressed in the following sections.

Many of the facial image processing techniques discussed in Chapter 2 made use of some form of image normalisation before further processing occured on the images. This could be to take into account variations in scale, position or rotation of the face within the image. These techniques were not required in this work as the original photographs had been taken under controlled conditions designed to eliminate such variations.

4.2.1 Colour representations

Colour representation[84] within digital images is possible in many different ways[19]. The most common method is RGB where three values are stored for each pixel giving the intensities of red, green and blue light for that pixel. This method is closely related to the hardware involved in the display of colour images which is normally produced using a combination of red, green and blue light. The range of variations in colour which are possible using this system can be represented using a colour cube as shown in Figure 4.2. Here one of the axis is taken as the red value, one as the green and the remaining one as the blue. All possible colours are then represented by the points within the cube with black at [0,0,0] and white at [1,1,1].

For colour printing, one of two methods is usually used, either CMY or CMYK where this time the intensities of cyan, magenta, yellow and black are measured. This is matched to the colours of the inks often used in combination to form the desired

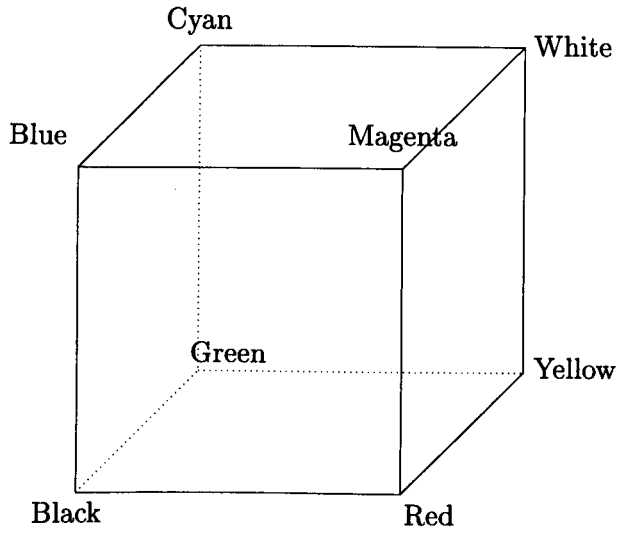


Figure 4.2 The RGB colour cube

colour. The pixel values in CMY can be found by taking each of the components in RGB and subtracting them from 1. An alternative scheme is HSV where the three components represent the hue, saturation and value or brightness of the pixel. The hue is the basic underlying colour of the pixel and the saturation tells how pure it is or, how much white is mixed with it. The HSV colour scheme can be represented by a cone as shown in Figure 4.3. The pure colours are around the edge of the top of the cone in a given order, hue is measured by an angle round this circle starting with red at 0°. Moving towards the centre of the cone adds white light to the colour and so saturation is the distance from the centreline of the cone. There is an algorithm for performing the conversion of RGB to HSV given in appendix B.

The HSV representation is widely used in image processing applications[25]. Considering the three components of the HSV representation, it becomes clear that this representation separates colour information from intensity (the *value* component). This gives significant advantages when image procecessing applications are being considered. For example, many effects of varying lighting conditions will alter primarily the intensity component, leaving the colour information unchanged. Image enhancement can be carried out using standard monochrome processing techniques such as histogram

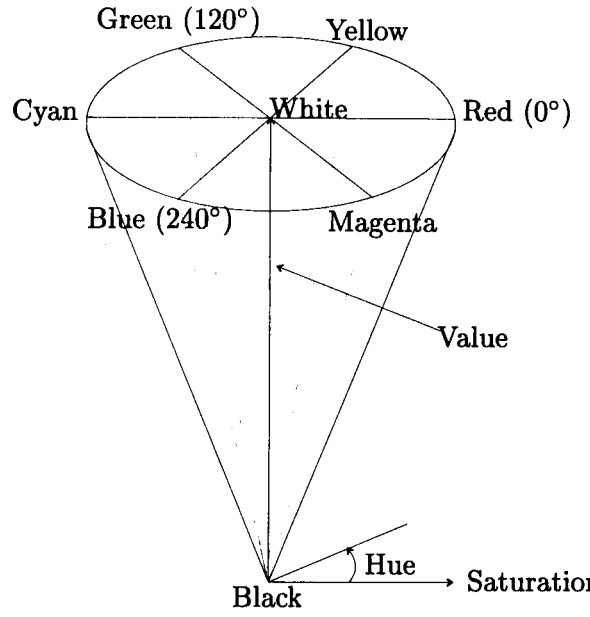


Figure 4.3 The HSV colour cone

equalisation on just the intensity component, resulting in an enhanced image without alteration to the colour properties. Alteration to the colour can be performed separately using appropriate algorithms. In addition, in this representation, the hue and saturation components bear close resemblance to the manner in which humans describe colours. I.e. the *hue* gives the underlying colour, such as orange, that a human would use in a description. The *saturation* then gives indication of how pure the colour is i.e. how much white light is mixed with the underlying orange colour. In this work, HSV, RGB and greyscale representations will be considered as methods of presenting the source data for analysis.

4.2.2 Image segmentation

We wish to optimise the performance of our classification system and one method of achieving this is to optimse the area of image presented to the network to include only that which is relevant to the classification in question. Thus eliminating data from other irrelevant features which may influence the network's performance. To implement this, use is made of the facial feature points that are associated with the data set. Many other people are working in the area of feature point location so in the work presented

here, it was assumed that the feature points would be available for any images to be processed.

With the supplied data set, the original set of points were unavailable so a new set were entered by the use of a program written by the author for the X window environment on Sun SPARC workstations. This process was less accurate than the original method using a projector and graphics tablet due to a reduction in image size once presented on the computer screen and the limitations in accuracy when using a mouse. Whereas the original system had a grid of 2500 × 2500 pixels covering the projected images, the X window based system was limited to the size of the sampled images, namely 384×512 pixels. However the original, higher resolution point locations would have made litte difference to the results as they would have had to be mapped onto the 384 × 512 pixel images.

The X window program written by the author served two processes. Its first was the gathering of feature point location data as described above. This was acheived by presenting the face images in turn on screen and using the mouse to point at the positions of the feature points of interest. These positions were stored in a file against the number of the face concerned. The system was also capable of displaying the faces with the features marked and listing the descriptive measures associated with that face.

The second main use of the program was the generation of the data sets described in Section 4.3. Routines were written to convert the TIFF images into suitable training data files for the neural network systems and other data processing systems used in analysis work. The user of the software could select any combination of colour representation, image area, sub-sampling and output format and the range of image numbers to include along with options about data set balancing, which descriptive measure the file refered to and what system it is to be used on. A facility was included to allow a batch of these requests to be processed sequentially which is how most of the data file sets were produced for this thesis.

The files produced by the system were checked by including the production of PPM[2] as one of the options. These PPM files were then viewed using standard X11

[2]PPM - Portable Pixel Map is a common simple graphics file format in the UNIX environment

Table 4.3 Feature points used in data location

Feature	Data point used	Approximate feature size
Moustache	30 – centre of top lip	100 × 30
Beard	32 – centre of bottom lip	200 × 70
Eye Colour	18 – centre of right eye	30 × 20

applications and compared with the original images to confirm that the correct area had been selected, the correct sub-sampling had been performed etc.

For each of the features that were considered, a point was chosen that was considered to be a suitable reference marker and then an image area was taken with respect to that marker. The reference markers were chosen as the feature point with the closest relationship to the feature in question. The set of feature points used for the various feature classifications are listed in Table 4.3. Having established a reference point for the areas of the images to be considered, suitable sized areas needed to be chosen. Consideration was given to the sizes of the features throughout the data set of faces and the maxima were found. The largest image size needed to enclose the features considered is also given in Table 4.3.

Even restricing data sets to the areas given in Table 4.3 will still result in the need for networks with several hundred inputs – a situation that could lead to difficulty in training the networks due to the number of weights that would be present in the system. So there is still a need reduce the number of elements in the input pattern vectors. One method of achieving this is to sub-sample the image and this method was employed by means of a simple pixel averaging process. When the sub-sampling was used, it was specified in terms of separate sub-sampling in the horizontal and vertical directions, for example an image sub-sampled at 2×5 would have a single pixel representing what was originally a 2×5 pixel area in the original image. Instead of the pixel averaging method used here, there are other methods that could have been used to reduce the size of the image. Pixel averaging will lose some of the detail in the image, however when considering the classification problems being tackled here, the detail of the image in terms of edges and fine lines is not the important part. Rather the classification systems have to detect overall properties of an area of the image.

Having said this, difficulties can still occur with information loss when sub-sampling RGB data. Since the three sets of data (R, G and B) are not linked mathematically when the sub-sampling occurs, the three final values are independant of each other. If adjacent pixels that are to be averaged have greatly differing colours and therefore differing RGB values, the process of pixel averaging can have the effect of producing a final pixel of a new colour that is not related to the colours of the original pixels. This problem is not found when sub-sampling HSV data as the colour information is contained in a single value for each pixel and therefore an "average" colour is produced in the resulting pixel. To evaluate the difficulty caused by this form of error, some of the RGB images that were sub-sampled were viewed on screen along with the original image. No discernable problems were detected as a result of the sub-sampling though it is possible that errors occured in images that were not examined.

The classification problems being tackled are presented next.

4.3 Data sets used

Two basic categories of features are being investigated in this work; some with a "yes/no" type measurement and some with a graded rating. The application of supervised neural networks to these problems is considered in Chapter 5 and Chapter 6 respectively. Two problems of the former type were examined, namely the identification of the presence or absence of moustaches and beards. The motivation for this was that we wished to explore the potential for neural network techniques in the area of facial feature classification. These "yes/no" classifications were regarded as "simple" cases and therefore usefull for evaluating the potential for neural networks in this work.

In each case a collection of ten data sets was used and the details of these are given in Table 4.4 for the moustaches and Table 4.5 for the beards. The data sets were designed to cover a wide range of variation in the size of image segment being considered and level of sub-sampling. In addition, each size of image was prepared with both greyscale and hue representations. While it is apparent to the human eye from a greyscale image whether a given face has moustache or beard, it was considered to be worthwhile to try an alternative colour representation in addition for the classification

Table 4.4 Data sets used for moustache identification

Name	Width	Height	Sub-sample	Colour	Elements
moustache01	100	5	5 × 5	grey	20
moustache06				hue	
moustache02	100	5	25 × 5	grey	4
moustache07				hue	
moustache03	100	25	5 × 25	grey	20
moustache08				hue	
moustache04	100	25	5 × 5	grey	100
moustache09				hue	
moustache05	100	25	25 × 5	grey	20
moustache10				hue	

Table 4.5 Data sets used for beard identification

Name	Width	Height	X offset	Y offset	Sub-sample	Colour	Elements
beard01	200	70	0	-35	20 × 7	hue	100
beard02						grey	
beard03	100	70	0	-35	20 × 7	hue	50
beard04						grey	
beard05	50	40	0	-50	10 × 10	hue	20
beard06						grey	
beard07	100	70	0	-35	40 × 7	grey	20
beard09						hue	
beard08	100	70	0	-35	25 × 14	grey	20
beard10						hue	

routines.

In the case of the graded classifications, one measure was selected for the investigations, namely eye colour. This was defined in terms of one of five colours as given in Table 4.6. Eye colour was chosen as a complex classification problem. Rather than being a simple scale of 1 to 5 as in the other 'graded' cases, representing some measure that could vary between two extremes, the eye colour case is represented as five distinct colours. These colours are not neccecarily related in the order in which they have been

Table 4.6 The eye colour classifications

Number	Colour description
1	blue
2	grey
3	green
4	hazel
5	brown

Table 4.7 Data sets used for eye colour identification

Name	Width	Height	X offset	Y offset	Sub-sample	Colour	Elements
eye01	20	7	0	0	1 × 1	hue	140
eye02						grey	
eye14						red	
eye15						green	
eye16						blue	
eye03	20	8	0	0	2 × 2	hue	40
eye04						grey	
eye17						red	
eye18						green	
eye19						blue	
eye05	8	8	5	0	1 × 1	hue	64
eye06						grey	
eye20						red	
eye21						green	
eye22						blue	
eye07	30	20	0	0	5 × 5	hue	24
eye08						grey	
eye23						red	
eye24						green	
eye25						blue	
eye09	8	8	5	0	2 × 2	red	16
eye10						green	
eye11						blue	
eye12						grey	
eye13						hue	

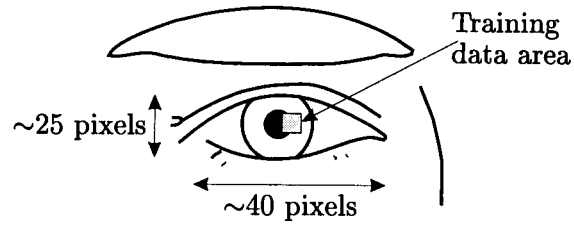


Figure 4.4 Offset data area for eye colour classification

aranged in the encoding of blue through to brown, therefore this case is different to all the other graded classifications.

The data sets used in the work on eye colour classification are detailed in Table 4.7. In this case, the use of alternative colour representations, in terms of the single components from the RGB representation, were also included. Since the desired classification was one of colour, it was envisaged that certain colour representations would highlight the differences in the colours more than others. Some of these data sets were designed to cover the whole eye while others contained only a horizontal window over its centre. This window would reduce the image size presented to the network but still include area containing the colour information. The 8×8 data sets are different in that they are arranged offset from the centre of the eye as shown in Figure 4.4, so as to avoid the pupil which is irrelevant for colour classification. In addition, these sections of image avoided some of the lighting reflections that were present in some of the photographs. The variation in size of image was used to investigate how the variation in quantity of data would affect the classification and whether the classification would benefit from considering only the coloured section of the eye.

With these data sets, the photographs of people wearing glasses were excluded as these often had bad reflections from the lighting, or the glasses were coloured in some manner, some to the extent that the eye could not properly be seen. Having eliminated these images, the number of patterns available in each class is given in Table 4.8

Table 4.8 Distribution of examples of each eye colour class

Eye colour	Training Patterns	Testing Patterns
Blue	486	57
Grey	40	7
Green	35	6
Hazel	56	4
Brown	263	22

4.4 Data analysis

Before applying supervised neural network techniques to learn the feature representations, the data was analysed using principal components analysis and also unsupervised neural networks to explore any natural groupings within the data. If clusters were identified using either of these techniques, which could be matched to the desired classifications, then this would indicate a greater probability of success when using the supervised neural network models. These clusters would show separability in the data which a neural network would be able to use in identifying the different classes contained within the data.

4.4.1 Principal Components Analysis

Principal Components Analysis[38] (PCA) is a technique for multivariate analysis. It is usually used with the objective of reducing the dimensionality of a data set which contains a number of interrelated variables, but at the same time keeping as much of the information as possible. This is done by transforming the original data into a new set of uncorrelated variables which are ordered such that the first few have most of the variation present in the original variables. The determination of the principal components is acheived via eigenvalues and eigenvectors as will now be explained.

Definition

Let $\mathbf{x}$ be a vector of n random variables. PCA will first determine a function $\boldsymbol{\phi}_1^\top \mathbf{x}$ which has maximum variance, where $\boldsymbol{\phi}_1 \triangleq [\phi_{11}\phi_{12}\ldots\phi_{1n}]$ such that

$$\boldsymbol{\phi}_1^\top \mathbf{x} = \phi_{11}x_1 + \phi_{12}x_2 + \cdots + \phi_{1n}x_n = \sum_{j=1}^{n} \phi_{1j}x_j \ . \tag{4.1}$$

Then a second linear function $\boldsymbol{\phi}_2^\top \mathbf{x}$ is determined which is uncorrelated to $\boldsymbol{\phi}_1^\top \mathbf{x}$ and still has maximum variance. This process is repeated such that at the jth stage, the linear function $\boldsymbol{\phi}_j^\top \mathbf{x}$ is found with maximum variance subject to being uncorrelated with $\boldsymbol{\phi}_1^\top \mathbf{x}, \ldots, \boldsymbol{\phi}_{j-1}^\top \mathbf{x}$. There may be up to n principal components (PCs) found, though generally most of the variation will be accounted for in m PCs, where $m << n$.

To calculate the PCs, consider the case where $\mathbf{x}$ has a known covariance matrix $\boldsymbol{\Sigma}$, i.e. the matrix whose (i,j)th element is the covariance between the ith and jth elements of $\mathbf{x}$ when $i \neq$j and the variance of the jth element when $i = j$. It can be shown that for $k = 1, 2, \ldots, n$, the kth PC is given by $z_k = \boldsymbol{\phi}_k^\top \mathbf{x}$ where $\boldsymbol{\phi}_k$ is an eigenvector of $\boldsymbol{\Sigma}$ corresponding to its kth largest eigenvalue λ_k. In addition, if $\boldsymbol{\phi}_k$ is chosen to have unit length then the variance of z_k is given by λ_k.

The following derivation of these results is a standard one found in many books[38]. First consider $\boldsymbol{\phi}_1^\top \mathbf{x}$. $\boldsymbol{\phi}_1$ by definition, maximises $var[\boldsymbol{\phi}_1^\top \mathbf{x}] = \boldsymbol{\phi}_1^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_1$. For this maximum to be achieved, a constraint must be imposed and a convenient one is $\boldsymbol{\phi}_1^\top \boldsymbol{\phi}_1 = 1$. Others may be used for particular circumstances and substituted later. Now to perform the maximisation, the technique of Lagrange multiplier is used so that

$$\boldsymbol{\phi}_1^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_1 - \lambda(\boldsymbol{\phi}_1^\top \boldsymbol{\phi}_1 - 1) \ , \tag{4.2}$$

is now the expression to be maximised where λ is the Lagrange multiplier. Differentiating (4.2) with respect to $\boldsymbol{\phi}_1$ gives

$$\boldsymbol{\Sigma} \boldsymbol{\phi}_1 - \lambda \boldsymbol{\phi}_1 = 0 \ , \tag{4.3}$$

or

$$(\mathbf{\Sigma} - \lambda \mathbf{I}_n)\boldsymbol{\phi}_1 = 0 \,, \tag{4.4}$$

where $\mathbf{I}_n$ is the $(n \times n)$ identity matrix. Therefore, λ is an eigenvalue of $\mathbf{\Sigma}$ and $\boldsymbol{\phi}_1$ is the corresponding eigenvector. To work out which of the n eigenvectors is the correct one for $\boldsymbol{\phi}_1$, consider that the expression to be maximised is

$$\boldsymbol{\phi}_1^\top \mathbf{\Sigma} \boldsymbol{\phi}_1 = \boldsymbol{\phi}_1^\top \lambda \boldsymbol{\phi}_1 = \lambda \boldsymbol{\phi}_1^\top \boldsymbol{\phi}_1 = \lambda \tag{4.5}$$

so λ needs to be as large as possible. Hence, $\boldsymbol{\phi}_1$ is the eigenvector corresponding to the largest eigenvalue of $\mathbf{\Sigma}$ and the variance of $\boldsymbol{\phi}_1^\top \mathbf{x}$ is $\boldsymbol{\phi}_1^\top \mathbf{\Sigma} \boldsymbol{\phi}_1 = \lambda_1$, i.e. the largest eigenvalue.

Now the second PC $\boldsymbol{\phi}_2 \mathbf{x}$, maximises $\boldsymbol{\phi}_2^\top \mathbf{\Sigma} \boldsymbol{\phi}_2$ with the constraint of being uncorrelated to $\boldsymbol{\phi}_1^\top \mathbf{x}$. If $\mathrm{cov}(x, y)$ denotes the covariance between the variables x, and y then this constraint is $\mathrm{cov}[\boldsymbol{\phi}_1^\top \mathbf{x}, \boldsymbol{\phi}_2^\top \mathbf{x}] = 0$. However

$$\mathrm{cov}[\boldsymbol{\phi}_1^\top \mathbf{x}, \boldsymbol{\phi}_2^\top \mathbf{x}] = \boldsymbol{\phi}_1^\top \mathbf{\Sigma} \boldsymbol{\phi}_2 = \boldsymbol{\phi}_2^\top \mathbf{\Sigma} \boldsymbol{\phi}_1 = \boldsymbol{\phi}_2^\top \lambda_1 \boldsymbol{\phi}_1 = \lambda_1 \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1 = \lambda_1 \boldsymbol{\phi}_1^\top \boldsymbol{\phi}_2 \,. \tag{4.6}$$

So

$$\boldsymbol{\phi}_1^\top \mathbf{\Sigma} \boldsymbol{\phi}_2 = 0 \,, \qquad \boldsymbol{\phi}_2^\top \mathbf{\Sigma} \boldsymbol{\phi}_1 = 0 \,,$$
$$\boldsymbol{\phi}_1^\top \boldsymbol{\phi}_2 = 0 \,, \qquad \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1 = 0 \,,$$

can all be used to give no correlation between $\boldsymbol{\phi}_1^\top \mathbf{x}$ and $\boldsymbol{\phi}_2^\top \mathbf{x}$. Making an arbitary choice of the last of these as the constraint on the maximisation and applying the same method as before, gives

$$\boldsymbol{\phi}_2^\top \mathbf{\Sigma} \boldsymbol{\phi}_2 - \lambda(\boldsymbol{\phi}_2^\top \boldsymbol{\phi}_2 - 1) - \gamma \boldsymbol{\phi}_2^\top \boldsymbol{\phi}_1 \,, \tag{4.7}$$

as the expression to be maximised, where λ and γ are Lagrange multipliers. Differentiating with respect to $\boldsymbol{\phi}_2$ gives

$$\mathbf{\Sigma} \boldsymbol{\phi}_2 - \lambda \boldsymbol{\phi}_2 - \gamma \boldsymbol{\phi}_1 = 0 \tag{4.8}$$

which can be multiplied by $\boldsymbol{\phi}_1^\top$ to give

$$\boldsymbol{\phi}_1^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_2 - \lambda \boldsymbol{\phi}_1^\top \boldsymbol{\phi}_2 - \gamma \boldsymbol{\phi}_1^\top \boldsymbol{\phi}_1 = 0 \; . \tag{4.9}$$

The first two terms of this are zero and $\boldsymbol{\phi}_1^\top \boldsymbol{\phi}_1 = 1$ so the expression reduces to $\gamma = 0$. Therefore

$$\boldsymbol{\Sigma} \boldsymbol{\phi}_2 - \lambda \boldsymbol{\phi}_2 = 0 \; , \tag{4.10}$$

or

$$(\boldsymbol{\Sigma} - \lambda \mathbf{I}_n) \boldsymbol{\phi}_2 = 0 \; , \tag{4.11}$$

so, again, λ is an eigenvalue of $\boldsymbol{\Sigma}$ with $\boldsymbol{\phi}_2$ as the corresponding eigenvector. Once again, it is $\lambda = \boldsymbol{\phi}_2^\top \boldsymbol{\Sigma} \boldsymbol{\phi}_2$ that is to be maximised so $\lambda = \lambda_2$ which is the second largest eigenvalue and $\boldsymbol{\phi}_2$ is the corresponding eigenvector.

This may be extended to the third, fourth, ..., nth PCs to show that $\boldsymbol{\phi}_3$, $\boldsymbol{\phi}_4$, ..., $\boldsymbol{\phi}_n$ are the eigenvectors of $\boldsymbol{\Sigma}$ corresponding to the third, fourth, ..., nth largest eigenvalue. In addition, the variance of the kth PC is λ_k.

Application to data

Discussion now follows of the results of applying PCA to our data sets.

Since PCA concentrates the variance in the original data into the first few components, any linear separability in the data is likely to be shown by plotting these components. In performing this analysis, a cut off point needed to be established to determine the number of principal components to be calculated. To give reasonable numbers of plots to consider, the following criteria were adopted. Principal components were calculated using a Matlab routine to evaluate sufficient components to include 80% of the variance in the original data, though no more than nine were calculated for any given data set in order to limit the quantity of data produced to a manageble level.

Matlab was chosen as the tool for this process as it has built in eigenvalue - eigenvector calculating routines and the author was familiar with its operation. The PCA routine written by the author evaluated the principal components from the data set under consideration. It summed the variance found in each of the components and

found n, the number of components required to include 80% of this variance, subject to the upper limit of 9 componenets.

These n components were then plotted in pairs, 1st against 2nd, 1st against 3rd, 1st against 4th, ... 1st against nth, 2nd against 3rd, ... 2nd against nth, ... $(n-1)$th against nth; the plots being examined for any clustering. The target value for each of the training vectors was used to determine the colour of the point plotted, thus enabling any groupings to be matched to the desired responses if possible. For printing in black and white, the coloured points have been converted to different shapes; in the beard/moustache identification cases, diamonds represent face images with moustaches or beards and the crosses are those without.

Moustache data

First let us consider the data sets used for the identification of moustaches, the PCA technique described above produced a total of 93 PCA plots for the ten different data sets examined. Two basic kinds of plots were observed here. Firstly, there were those that showed no form of clustering. In these all the data points formed a single group, with no discernable separation; the data points belonging to the two classes were mixed up with each other, Figure 4.5(a) shows a typical example of this. Around 70% of the plots took another form and showed clustering that could be attributed to the two different classes of data being presented. The degree to which the two classes could be separated was variable, though a good example is the plot shown in Figure 4.5(b). From looking at Table 4.4, it is clear that all the data sets were taken from similar areas of the original images and therefore might be expected to contain similar information. This is demonstrated in the PCA plots by the fact that plots of the same pair of components for different data sets have the same basic shape. A good demonstration of this is seen by considering the plots of the first two principal components of data sets 06, 08, 09 and 10. These are shown in Figure 4.5(c) to Figure 4.5(f) and the same distinctive shape can be seen in all four. Other groups of plots share the same shape, but this is the most distinctive set. The plots shown in Figure 4.5 are only a selection of those that were examined in applying this technique to the moustache data and were found

to be representative.

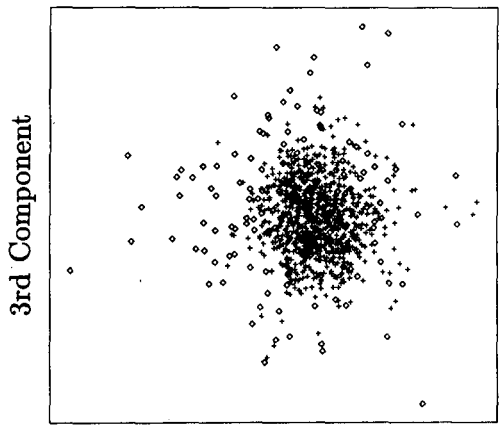
In none of the cases here is the distinction between the classes clear enough for a linear classifier to give perfect results using two principal components as the source data. It may still be possible for a linear classifier to be used with more than two of the components as input, however it is also possible that this problem is not linearly solvable. In Chapter 5 work is presented which is concerned with exploring the effectiveness of the inherently non-linear nature of some ANNs in the solution of this problem.

Beard data

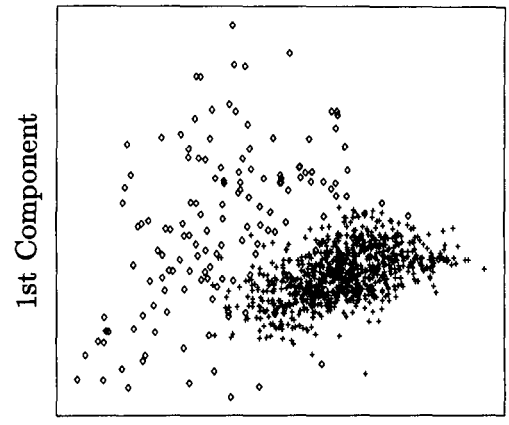
Applying the same criteria to the beard data for selecting the principal components to plot, resulted in a total set of 166 plots for the ten data sets, two typical examples are shown in Figure 4.6. In this case, around 80% of the plots showed little or no clustering as in Figure 4.6(a). However, there were some that showed clustering that had a reasonable match with the desired target cases as demonstrated by Figure 4.6(b) though the distinction was not as clear as in the case of the moustache data. Again, this shows that there is separability in the data set and indicates that a neural network should be capable of producing a solution in this case.

Eye colour data

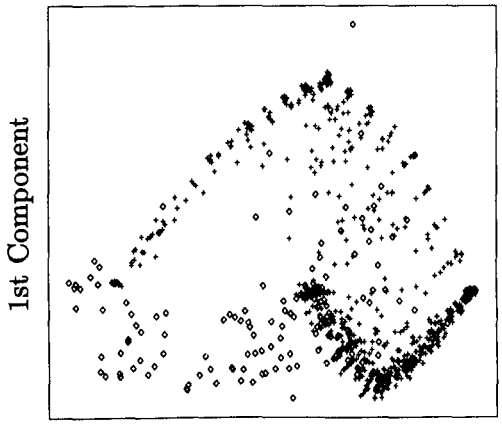
Turning now to the case of the eye colour data, here there are five classes so in the plots, five different shapes are used to indicate the classes represented by each of the points. These are listed against the colours that they represent in Table 4.9. Using the same rules for the number of principal components to be evaluated and the plotting of the components as described previously, resulted in a total of 213 plots covering the thirteen data sets used for this work. These plots typically showed very little in the way of clustering as indicated by the sample plots shown in Figure 4.7. Some had a single area of the plot where there was a dense concentration of points and then the remainder scattered round the edge, while others had a more even distribution of the points. However, no clustering was observed in any of the plots produced by this



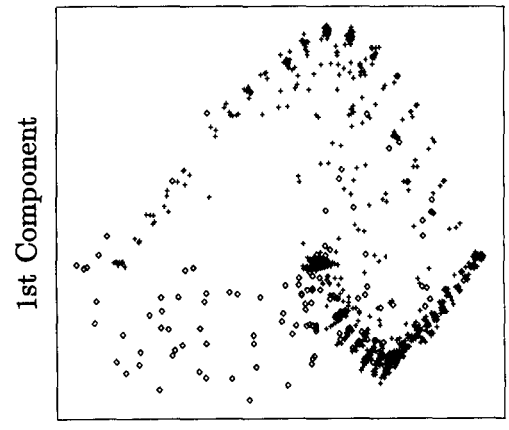
(a) No clustering (data set 01)



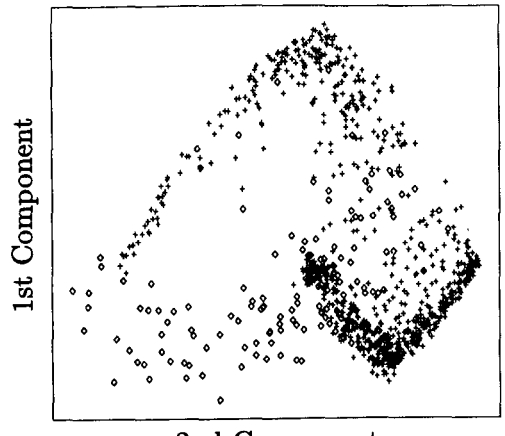
(b) Clustering according to data classifications (data set 01)



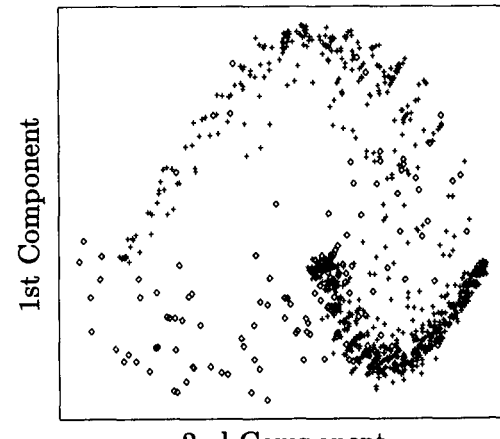
(c) data set 06



(d) data set 08

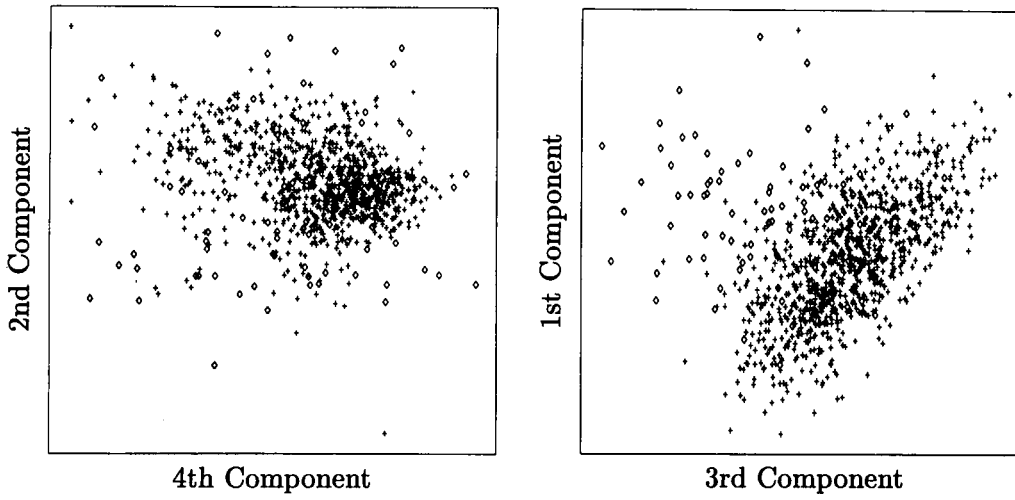


(e) data set 09



(f) data set 10

Figure 4.5 PCA plots using the moustache data sets



(a) No clustering (data set 02)

(b) Clustering according to data classifications (data set 07)

Figure 4.6 PCA plots using the beard data sets

Table 4.9 Shapes used to represent eye colours in PCA plots

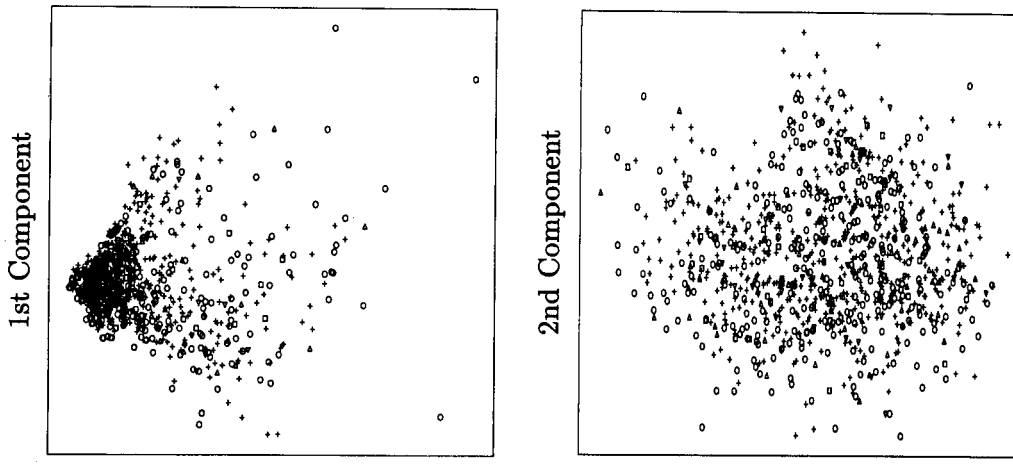
Colour description	Shape
blue	+
grey	□
green	▽
hazel	△
brown	○

method.

With the lack of clustering produced in the PCA plots, we can say that the classification of eye colour is not possible using linear separation given the data sets used. This was an expected result due to the complex nature of colour information. Kohonen Self-Organising Maps were then applied to the all data to investigate the use of a non-linear clustering technique.

4.4.2 Unsupervised Neural network methods

The Kohonen network and its training algorithm is described in Section 3.6.1. Various sizes of Kohonen SOM maps were applied to the data sets as a means of evaluating a



(a) Data set 01 (b) Data set 02

Figure 4.7 PCA plots using the eye data sets

non-linear clustering algorithm. All the maps were produced using the self-organising map algorithm supplied with the Neural Works package.

Neural Works was chosen as the simulation tool for all the artificial neural network simulations performed in this thesis. One consideration when selecting the package to use was that of being able to perform all the simulations in a single system, thus eliminating any differences caused by variations in the implementation of the network functions. Neural Works has an extensive range of network architectures and learning algorithms built in. In addition, it is able to run a group of simulations in "batch" mode allowing unattended simulation runs to be performed - essential for the number of simulations that were performed as part of this thesis work. Data file and results file formats were simple to process using various data manipulation tools, and while the generation of network architectures cannot be automated, the author developed methods for replicating identical network architectures that were used on different data sets. Various packages were considered before a choice was made, but from the collection of packages available to the author, Neural Works was considered to be the most appropriate for the task.

Larger SOM maps provide more space for clusters to form though are more compu-

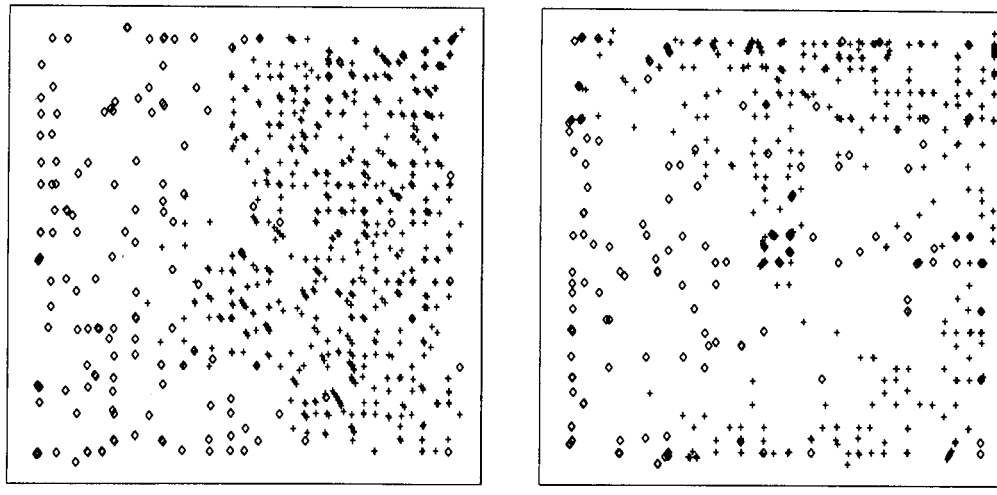
tation intensive in operation due to the increased number of neurons. For this reason, often the procedure applied to find a good SOM network for clustering a particular data set is to start with a large map and reduce the size until the network is no longer able to cluster the data. One indication that the map is larger than it needs to be is that the data does not cover the entire area of the map, i.e. there are neurons in the map which are not being used.

The networks used had a "co-ordinate" output layer of two neurons. The values on these two neurons gave the co-ordinates of the winning neuron within the map and these co-ordinate values were plotted, using different coloured points to denote the different classes within the data. The output also has an option to interpolate the co-ordinate values of the three neurons producing the strongest signal and this was used in this analysis. This gives the plots the appearance that there are more neurons in the map than there actually were as co-ordinate values *between* the neurons can be generated.

Moustache data

The principle component analysis has already revealed separability within the moustache data set into the two classes. Now the Kohonen SOM provides an alternative clustering method to confirm this. Two of the plots produced from the moustache data sets are shown in Figure 4.8. Figure 4.8(a) shows a typical plot produced using one of the greyscale data sets. A clear distinction can be seen between the two classes in the data set with the diamonds representing faces with moustaches to the left and the crosses representing those without moustaches to the right.

Figure 4.8(b) shows the Kohonen plot produced using the same size map with the equivalent data set using hue data rather than greyscale. The degree of separation into the two classes that was observed in Figure 4.8(a) is not seen here. These plots demonstrate that information contained within the greyscale data naturally groups the vectors according to the two classes of interest whereas the hue information does not produce this clustering. This does not mean that the hue data cannot be used to perform this separation, rather it means that the predominant grouping within that data set is not the one which is being sought after here.



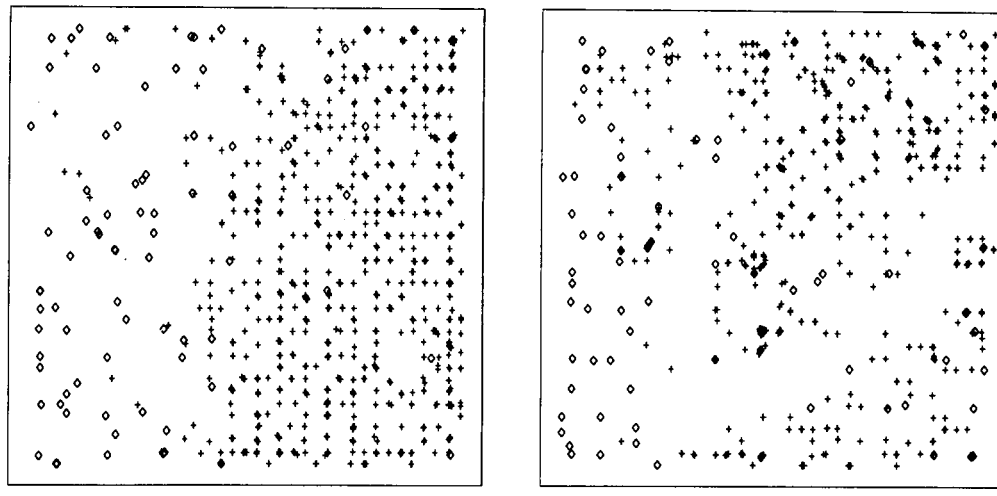
(a) Data set 01 (b) Data set 06

Figure 4.8 SOM plots using the moustache data sets

Beard data

The same form of map is observed in Figure 4.9 when considering the beard data sets. Figure 4.9(a) shows a SOM generated using a greyscale data set. As with the greyscale data for moustache classification, there is a clear distinction between the two classes in the data set. It is coincidence that both of these plots show the classes separated in the same manner. The arrangement of clusters within a SOM plot is dependent on the original random weights that are used to initialise the network; other plots of the same data have the clusters in different arrangements, e.g. some have a "top/bottom" split instead of the "left/right" split seen here.

Figure 4.9(b) is the SOM plot produced using the hue representation of the same data set. It can be seen that, while there is some clustering in the data it has not separated according to the classes of interest. As with the hue data for moustache classification, this does not mean that this data cannot be used for such purposes, rather, it shows that the primary clustering is not according to the classes being examined in this instance.



(a) Data set 07 (b) Data set 09

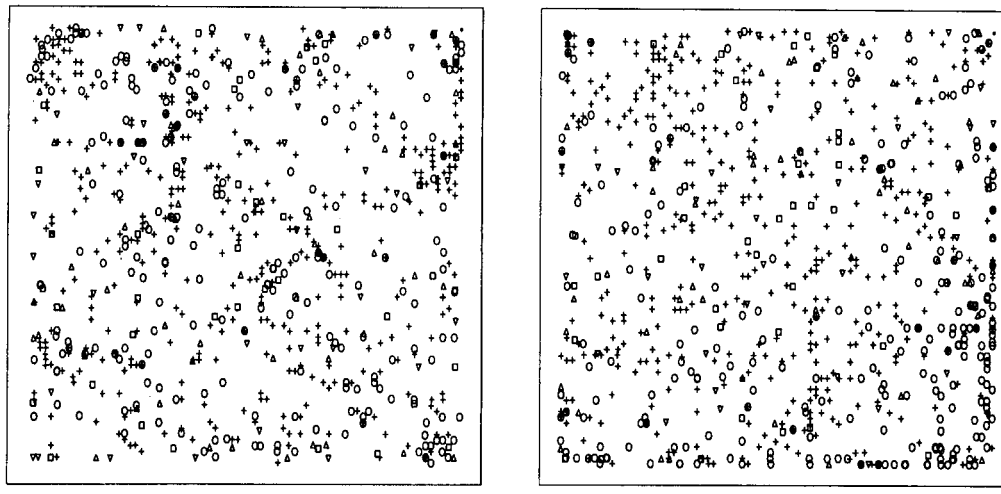
Figure 4.9 SOM plots using the beard data sets

Eye colour data

When the PCA algorithm was applied to the data used for eye colour classification (see Section 4.4.1), it was clear that the classes of eye colour were not linearly separable. Now using these data sets to train various sizes of Kohonen SOM will reveal whether a non-linear clustering technique will identify the five eye colours as separable classes.

The same five shapes are used in Figure 4.10 to represent the five different eye colours as were used in the PCA plots produced from the eye colour data (see Figure 4.7 and Table 4.9). Figure 4.10 shows two typical SOM plots that were obtained by this process, one for each of the first two data sets. The SOM networks used in producing these plots of the eye colour data were 20×20 in size, thus having a total of 400 neurons in the SOM layer. Both plots fail to show any form of clustering according to the five eye colours. Figure 4.10(a) does show some element of clustering within the data but this does not match the classes that are being examined here.

What is shown here is that these data sets do not naturally separate according to the five eye colours that have been defined. It would be the subject of an entirely different study to investigate what clustering is possible with this data but it may be concluded that automated classification according to the eye colours is not straight



(a) Data set 01 (b) Data set 02

Figure 4.10 SOM plots using the eye data sets

forward.

4.5 Conclusions

A number of different data sets have been produced in order to perform the classification of three different facial features, namely, moustache, beard and eye colour. There are a huge number of variations of the parameters that are possible in the production of the data sets used in this work. The ones produced here have been designed to cover a large range of different combinations of image size and, in the case of eye colour, colour representation.

Initial analysis of the data sets using principal component analysis and Kohonen self-organising maps have revealed the potential for separation in the data sets describing moustache and beard features and therefore the potential for automatic classification of these features. The eye colour data sets, however, have shown no significant signs of separability with the two techniques used here. This does not rule out the possibility of automatically classifying eye colour based on the information available but does show that it is a complex task.

The subsequent chapters will examine how well supervised neural networks perform

in terms of realising these classifications.

Chapter 5

Simple Feature Classification

The descriptive measures of faces which for the basis of the work in this thesis may be considered to be of one of two types, either binary or multi-valued (see Chapter 4). This chapter considers the classification of some of the former cases, namely, identification of the presence of beards and moustaches.

5.1 Experimental Method

The data sets listed in Table 4.4 and Table 4.5 are used in these experiments. In both cases, let class $\mathcal{A}$ be the set of faces possessing the feature in question (moustache or beard) and class $\mathcal{B}$ be the set without the given feature. As may be expected with a dataset representative of a subsection of the population, in both instances, class $\mathcal{A}$ has less members than class $\mathcal{B}$. This can cause difficulties in some classifier systems, producing a bias towards class $\mathcal{B}$, hence the data sets used in these experiments were balanced by including multiple copies of members of class $\mathcal{A}$ such that the number of occurances of members of class $\mathcal{A}$ and class $\mathcal{B}$ in the data sets is equal. Algorithm 5.1 extends this principle to the balancing operation in the case where there are an arbitrary number of classes $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, ... in the data set.

For all experiments, the data sets were split into two sections, one containing sections from 900 images is used as the *training* data and the remaining 100 images are used to form the *testing* data. Some images in the data set had dark shadows near the bottom of the image, possibly an error introduced at the scanning stage of data

Algorithm 5.1 Evaluate the repetition needed of each class to balance data sets

$n_{\mathcal{A}} \leftarrow$ number of examples of class $\mathcal{A}$
$n_{\mathcal{B}} \leftarrow$ number of examples of class $\mathcal{B}$
$n_{\mathcal{C}} \leftarrow$ number of examples of class $\mathcal{C}$ ...

$\mathbf{n} \triangleq [n_{\mathcal{A}}, n_{\mathcal{B}}, n_{\mathcal{C}}, \ldots]$

$most \leftarrow$ MAX(n)

for all n_i where $i = \mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$ **do**
  $r_i \leftarrow$ INT$((most/n_i)+0.5)$
end for

Table 5.1 Data set sizes for simple classifications

Data set	training patterns	testing patterns
Moustache	1504	120
Beard	1555	235

capture. In some cases this disguised the chin in such a way that a beard or moustache would not be identifiable so for these experiments the images with such shadows were omitted from the data sets. The balancing algorithm was applied independently to both testing and training data sets resulting in training and testing data sets of the sizes given in Table 5.1. The differing sizes of the moustache and beard data sets was caused by there being a smaller number of images with beards present than those with moustaches therefore the balancing routine needed to multiply the presentation of the images with beards more than those with moustaches.

5.1.1 Statistical classifiers

In order to evaluate any benefit of applying neural network techniques to these classification problems, it is necessary to investigate the use of some "conventional" classification techniques. Two different methods are considered here, linear regression and distance measures.

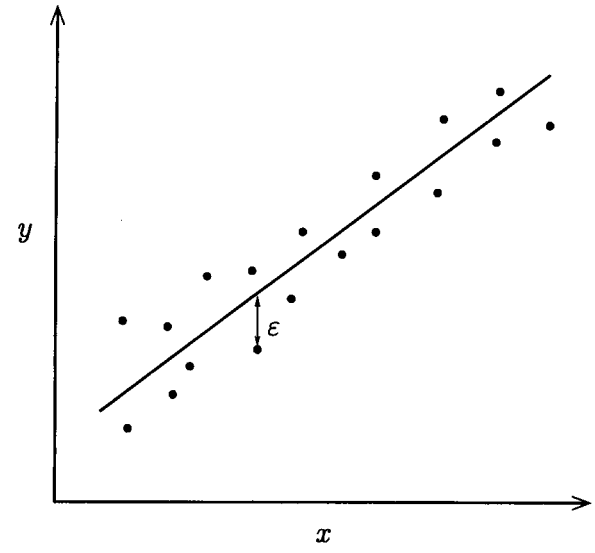


Figure 5.1 Simple linear regression

Linear Regression Models

Simple linear regression fits a function of the form

$$y = \beta_0 + \beta_1 x \tag{5.1}$$

to a set of data points by finding the parameters β_0 and β_1 such that the sum square of the errors, ε, between the data points and the y values returned by (5.1) is minimised. An example of this is shown in Figure 5.1. The data used in these experiments, however, has a vector as the input, $\mathbf{x}$, rather than a single variable. For this, multiple linear regression may be used. This technique extends the simple case to fit a function of the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n \ , \tag{5.2}$$

to the data by a least squares error technique. This method may be applied to these simple classifications since the classification may be represented by a single output, y, as a function of the input data, $\mathbf{x}$. The output classification may be defined such that class $\mathcal{A}$ is represented by an output value of 1 and class $\mathcal{B}$ by an output value of -1.

More precisely, the data is modelled by

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{5.3}$$

where $\mathbf{y}$ is the $(n \times 1)$ vector of desired classifications, $\mathbf{X}$ is the $(n \times p)$ matrix of input vectors, each having a 1 as the first element, $\boldsymbol{\beta}$ is the $(p \times 1)$ vector of regression coefficients and $\boldsymbol{\epsilon}$ is the $(n \times 1)$ vector of errors introduced by the regression model. The least squares technique is applied to minimise $\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}$ and results in the least squares estimator of $\widehat{\boldsymbol{\beta}}$ being evaluated as

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \; . \tag{5.4}$$

A more detailed explanation of this technique may be found in [56, Chapter 4].

The linear regression algorithm used in this thesis was implemented in Matlab. As the calculation of $\widehat{\boldsymbol{\beta}}$ is simple matrix manupulation, Matlab is an ideal candidate for this work with the algorithm implemented in a few lines of code.

In applying linear regression to the moustache and beard identification problems, the training data sets were used according to (5.4) to produce regression models. The estimated classifications, $\widehat{\mathbf{y}}$, were then calculated as

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}} \; , \tag{5.5}$$

where $\mathbf{X}$ was the input data from either the training or testing data sets. The classifications from $\widehat{\mathbf{y}}$ were then compared with the known true classifications, $\mathbf{y}$, and measures of accuracy calculated. This method was followed for all ten data sets for both the moustache and beard classification problems. A threshold was applied to the classifications made using (5.5) such that outputs with a magnitude of less than 0.2 were considered to be unclassified whereas those greater than 0.2 were taken as members of class $\mathcal{A}$ and those less than -0.2 as members of class $\mathcal{B}$. The same technique was used for the neural network methods as detailed in Section 5.1.2. See Section 5.2 for a discussion of the results.

Distance Classifiers

The training data is split into the two classes and the mean of each is calculated. To perform the classification, a distance measure is taken between the vector to be classified and each of the class means, and the one with the shortest distance is taken to be the correct classification.

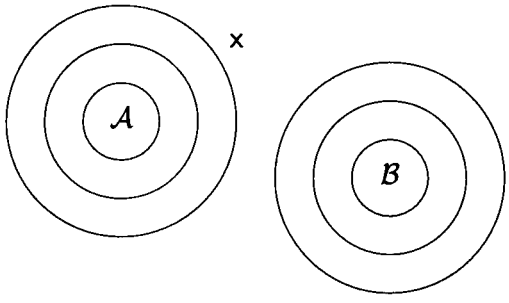
A general distance formula

$$d_i = (\mathbf{x} - \mathbf{c}_i)\mathbf{M}(\mathbf{x} - \mathbf{c}_i)^\top \tag{5.6}$$

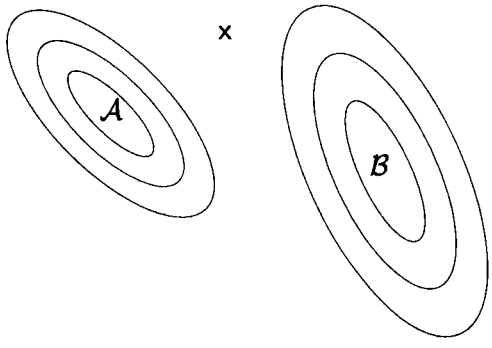
is used and by the multiplier matrix $\mathbf{M}$ the form of the distance measure is changed. d_i is the distance of input vector $\mathbf{x}$ from the centre, $\mathbf{c}_i$, of class i. With $\mathbf{M}$ set to $\mathbf{I}$, the identity matrix, the measure used is the Euclidean distance.

An alternative method sets $\mathbf{M}$ to the inverse of the covariance matrix of the training data for the class under consideration. This leads to distance measures that reflect the distribution of the data within the input space. Considering a 2D input space, the Euclidean distance may be thought of as a radius measure from the centre of the class. Using the inverse covariance matrix alters the measures such that the radius is now based on an elliptical co-ordinate system which will reflect the distribution of the data for the class within the input space. This is more clearly seen by considering Figure 5.2. In both diagrams, the centres of classes $\mathcal{A}$ and $\mathcal{B}$ are in the same position and the location of the unknown vector x that requires classification is also constant. In Figure 5.2(a), distances as measured using the Euclidean distance metric are represented by the circles surrounding the class centres. Under this distance metric, x is measured as being closer to $\mathcal{A}$ than $\mathcal{B}$. In Figure 5.2(b), the distance measures are now represented by a series of ellipses, representative of the distribution of the class data. Using this distance metric, the unknown, x, is now closer to $\mathcal{B}$ than to $\mathcal{A}$.

When using these distance classifiers, the resulting classification is simply the class which the unknown vector is closest to. Unlike the linear regression and neural network approaches, there is no set range for the output values of a distance metric. This means that a threshold cannot be set here to have an equivalent meaning to the thresholds



(a) Euclidean



(b) $\mathbf{M} = \mathbf{S}_i^{-1}$

Figure 5.2 Effect of different distance metrics

used in the other techniques. For this reason, the classifications made in this thesis using distance measures do not have the possibility of returning an *undetermined* state unless the distances given by the measures are equal. See Section 5.2 for a discussion of the results.

5.1.2 MLP Networks

The architecture of a multi-layer perceptron is presented in section 3.3.1 and the back-propagation of error method of training such networks in section 3.4.4. These are the techniques used here as a neural network solution to the problem of identifying moustaches and beards. With the backpropagation learning algorithm, a decision must be made as to when the network is 'trained'. This is based on some measurable parameter taken from the network and the appropriate choice is dependent on the problem being investigated. Two such measures, namely *r.m.s. error* and *accuracy*, are discussed in section 3.5. Since the goal of this system is a good rate of classification, the accuracy measure is the more appropriate to use here. The learning scheme followed is given in Algorithm 5.2. This method is designed to improve the generalisation ability of the network by the use of two data files. If decision as to when to stop the training of the network were based only on the accuracy achieved with the training file, it is likely that the network would achieve a good mapping for the data in that file possibly to the exclusion of data outside the training set. This is generally referred to as *over-training.*

In a typical network, as training progresses, the weights adjust so that the network more accurately reflects the mapping contained within the training data. In terms of the measures used to quantify the network's performance as presented in section 3.5, the training process will reduce the r.m.s. error when presented with training data. The degree to which this reduces the r.m.s. error of the network when testing data is presented, will depend on the correlation between the two data sets. If r.m.s. error is plotted as a function of training iteration then the resulting graph will usually be of the form shown in Figure 5.3. This shows how the error from the training data decreases through out the learning process while the error from the testing data initially

Algorithm 5.2 Learning scheme used for classification problems

load training data
load testing data

oldacc $\leftarrow$ 0 {to store the highest accuracy achieved}
repeat
  train for n iterations using training data
  calculate accuracy, *acc*, using testing data
  if *acc* $>$ *oldacc* **then**
    save the current set of the network
    oldacc $\leftarrow$ *acc*
  end if
until $R \times n$ iterations of training since the last network save

trained network is last version saved

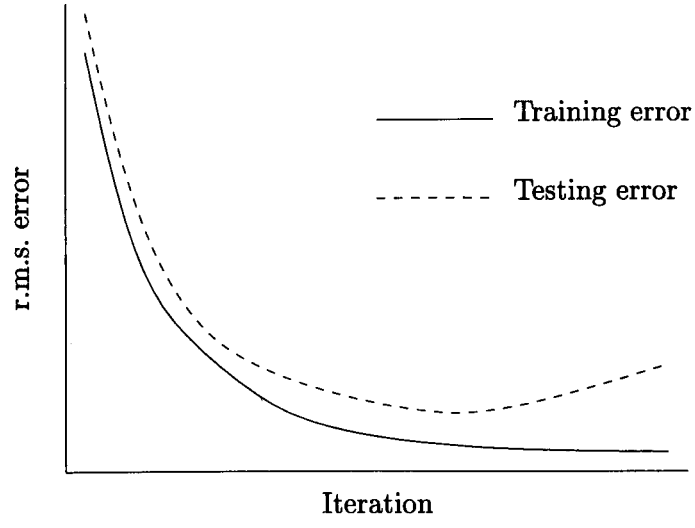


Figure 5.3 Typical error curves through the training cycle

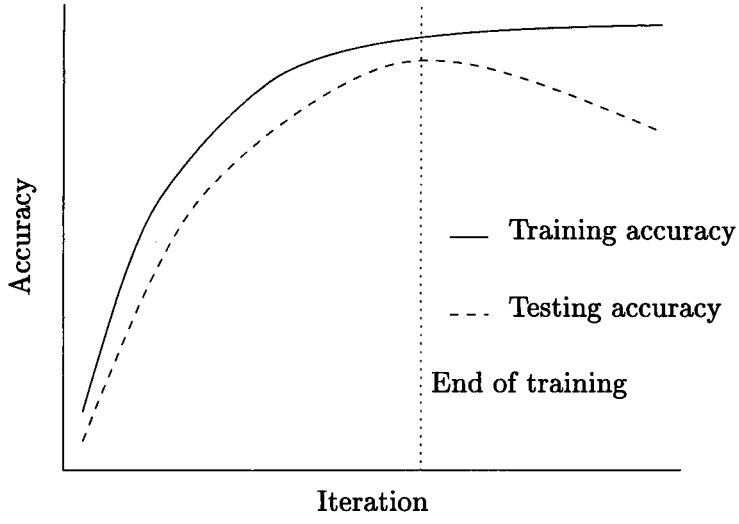


Figure 5.4 Typical accuracy curves through the training cycle

decreases but then rises again as the network mapping begins to match the training data too closely. In order to achieve best generalisation, learning schemes such as the one given in Algorithm 5.2 are used. In this case it is the classification accuracy that is considered rather than the r.m.s. error. Typical graphs of accuracy against iteration are represented in Figure 5.4. Following the learning scheme in Algorithm 5.2, training will stop at the point marked by the vertical dotted line where the testing accuracy is at a maximum. Note that the curves shown in Figure 5.3 and Figure 5.4 are representative of "well behaved" data where the distribution of both the training and testing data set within pattern space is similar. If this is not the case then the relationship between the training and testing accuracy will be different. It may be the case for example that the testing accuracy at some points during training is higher than the training accuracy; or if the two data sets are significantly differernt in terms of the required function to map input to output, then the testing accuracy will fall as the training accuracy increases.

Practically, the network simulations in this work were run using NeuralWorks II+[58] on Sun SPARC workstations. For each network configuration, the simulation was run five times with different initial conditions. This approach reduced the possibility that results in any given configuration were more due to the initial weights than to the network configuration. Five training runs were performed as a practical number

that could be achieved in the time available. The results used were the best from each of the groups of five simulations. This work followed on from some earlier simulations performed by the author using the Aspirin neural network simulator[62]. The investigations in this earlier stage of the author's work revealed that neural networks trained for these problems need small learning rates so the values used in this current study using NerualWorks were set to reflect this. See Section 5.2 for a discussion of the results.

5.1.3 Radial Basis Function networks

Radial Basis Function networks are presented in section 3.3.2 and the associated training algorithm in section 3.4.5. As implemented in NeuralWorks, this is a two stage process where firstly the hidden layer is trained for a number of iterations based on the size of the data set and then the output layer is trained in the same manner as for the MLP using the backpropagation of error technique. Algorithm 5.2 was once more used to set the stopping criteria for the training the output layer. In these simulations, for each data set, five different sized basis function layers were used, each trained using five different learning rates on the output layer. In the same way as for the MLP simulations, each combination was repeated five times, with different initial random conditions, in order to reduce the effect of these conditions on the final results. The results are discused in the following section.

5.2 Results

Throughout the results section of this chapter, reference is made to the various data sets defined in section 4.3. As defined, the data set names for the problems considered here are of the form moustache*nn* or beard*nn*. These will be abbreviated to m*nn* and b*nn* respectively for the purposes of this section. It should also be noted that the data sets are not presented in strict numerical order according to their name. This is in order to facilitate grouping by certain common features, e.g. colour type used in generating the data (greyscale / hue).

When examining the 'percentage accuracy' results, it should be born in mind that in two class cases such as those examined here, a random classification will yield 50%

accuracy. In other words, if the classification were performed using a random number generator, its accuracy would be 50% provided that there are equal numbers of examples of each class. However, this kind of classification has no form of confidence measure and the threshold term used in linear regression and neural network classifiers have no meaning. These classifiers can be most accurately compared with the random condition if the threshold term is not used.

5.2.1 Moustache classification

Consideration will be given to the two classification problems in turn; firstly the case of identifying the presence of moustaches. Results for the different techniques are presented in the same order as the methods already discussed in this chapter.

Statistical classifiers

The three statistical classifiers discussed in section 5.1.1 were applied to the ten moustache identification data sets. The results from the linear regression model are shown in Figure 5.5. The data sets may be split into two groups, m01–m05 use greyscale data whereas m06–m10 use hue data. Firstly considering the greyscale data sets, it can be seen that the highest accuracy is achieved with data set m04 and the lowest with data set m02. These are the data sets with the greatest and smallest number of elements per training vector respectively. Each of the others gives similar accuracies and these all have the same number of elements per vector. With the linear regression model, the number of parameters in the model is equal to the number of elements in the vectors plus one, showing that the models become more accurate as the number of parameters increases. Accuracy values of between 72% and 95% show that the grey scale data can be used in forming a linear model to classify the presence or absence of moustaches.

Turning to data sets m06–m10, it can be seen that the accuracies achieved here are much lower and also there is a greater difference between the accuracy for the training and testing data in any given data set. This indicates that the relationship between the hue colour data and the desired classification is not a linear one and this type of model is not suitable as a classifier for these data sets.

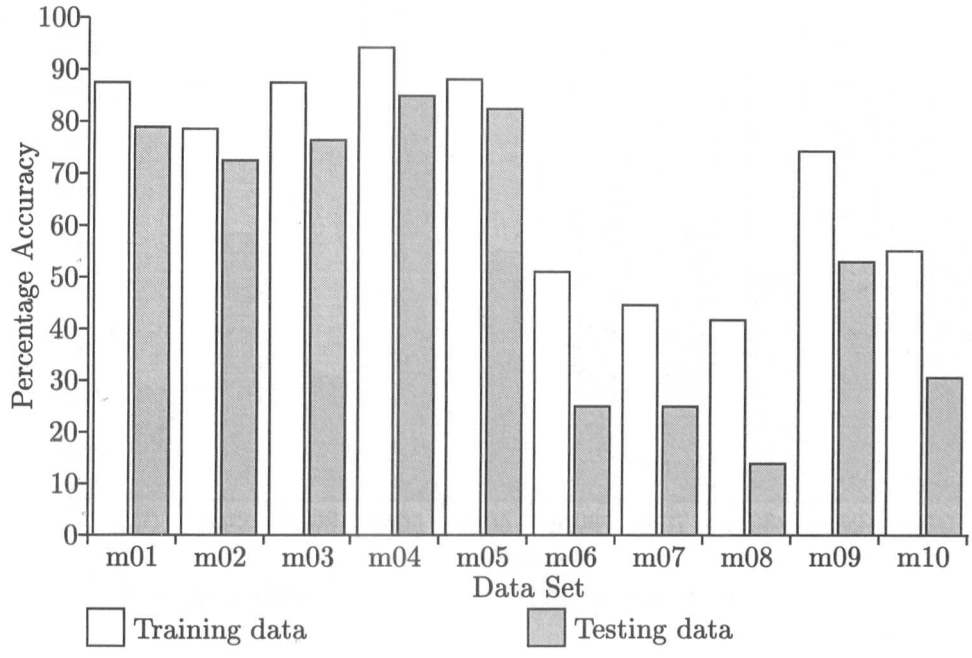


Figure 5.5 Linear regression classification results for moustache data sets

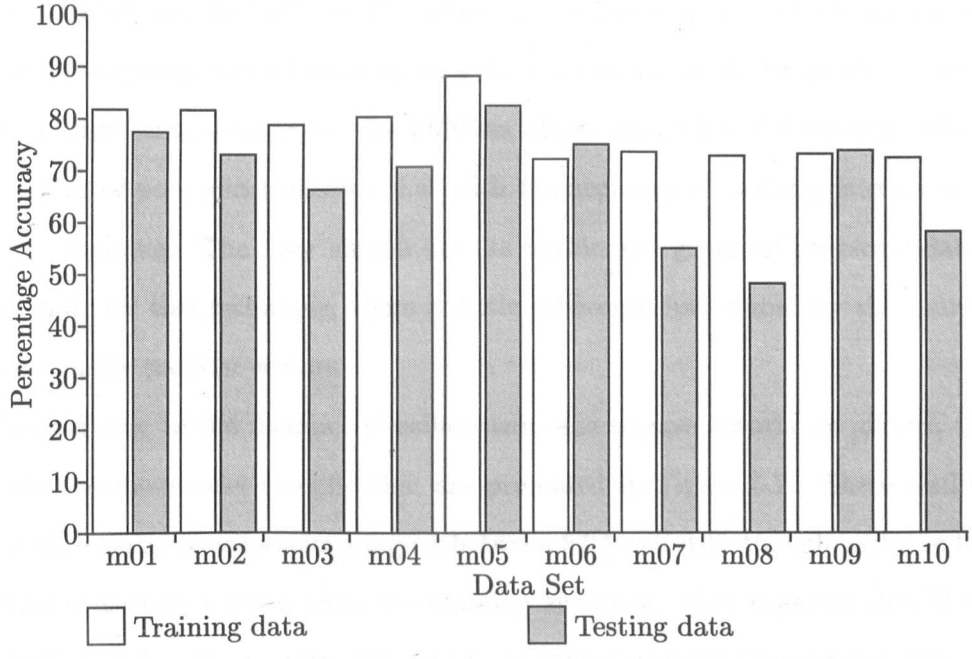


Figure 5.6 Euclidean distance classification results for moustache data sets

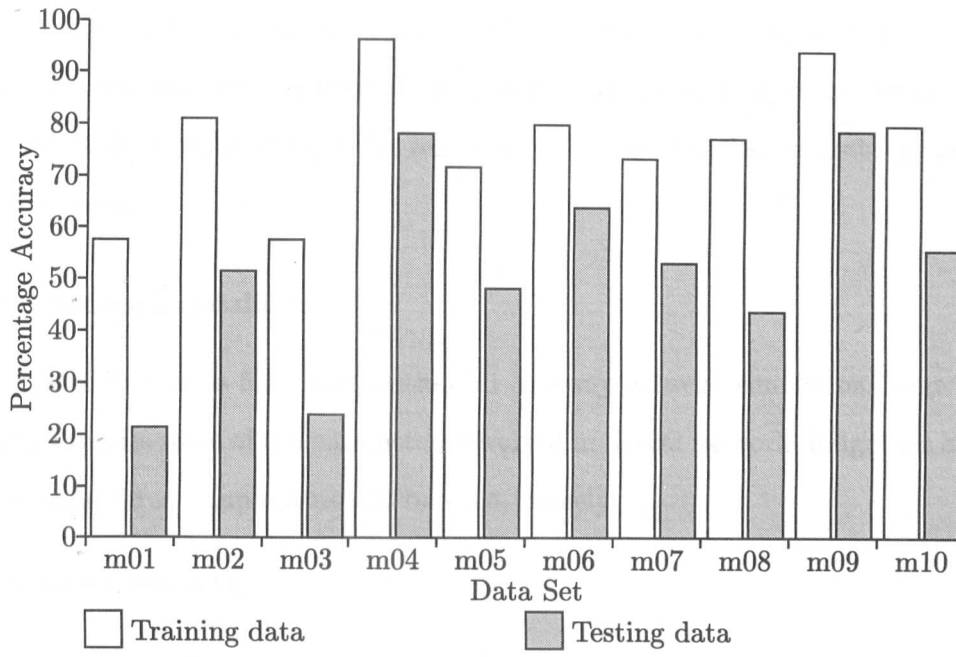


Figure 5.7 $\mathbf{M} = \mathbf{S}_i^{-1}$ distance classification results for moustache data sets

Results for the Euclidean distance classifier applied to the moustache data sets are shown in Figure 5.6. As in the previous case, a clear distinction can be seen between the data sets using greyscale information and those using hue with the greyscale returning a higher classification accuracy. In addition, three of hue based data sets, m07, m08 and m10 show poor generalisation, i.e. with the accuracy on testing data much lower than the training. The very similar results within the group of greyscale data sets shows that, for this technique, there is little difference introduced by the number of elements in the pattern vectors.

Turning now to the distance classifier using the inverse covariance matrix, the results for the moustache classification are presented in Figure 5.7. These results are erratic with training accuracies ranging between 57% and 94% though with the testing accuracy consistently lower than the training accuracy. This suggests that the data distribution within the pattern space does not match well that assumed by this model and also the distribution within the testing data set does not match that of the training data set.

The high accuracies that have been achieved with linear statistical classifiers are in line with the expectations from the principal components analysis presented in section 4.4.1 where certain plots on the moustache data set revealed separability between the two classes.

Neural network classifiers

As indicated in section 5.1.2 and section 5.1.3, many network simulations were run in order to optimise some of the parameters involved in neural network design. As a result the following three comparisons will be used, namely,

- between data sets,
- between learning rates,
- between the number of hidden neurons (MLP) or the number of basis function neurons (RBFN).

Some insight has already been gained into the differences between the ten data sets by use of PCA and the statistical classifiers. The other two comparisons are comparisons that are only relevant for neural systems and so there is no prediction of behaviour that can be gained from the work so far. However, having achieved high accuracies using linear classifiers, one would expect that a MLP system with zero hidden neurons would produce good results as this is also a linear system.

In total, for each data set, a set of eight different MLP network topologies were simulated with each of five different learning rates and each was simulated five times. In addition five different RBFN topologies were simulated with five different learning rates, each simulated five times. This gave a total of 3250 network simulations. Naturally, this number of results is too many from which to draw a meaningful comparison to be made here so examples are given to show how the above comparisons may be made.

Firstly consideration will be given to the multilayer perceptron networks. From the statistical classification results, it can be seen that the greyscale data sets achieve greater accuracy than the hue data sets. This trend is continued with the MLP classifiers. Figure 5.8 presents a comparison of classification accuracies using the different

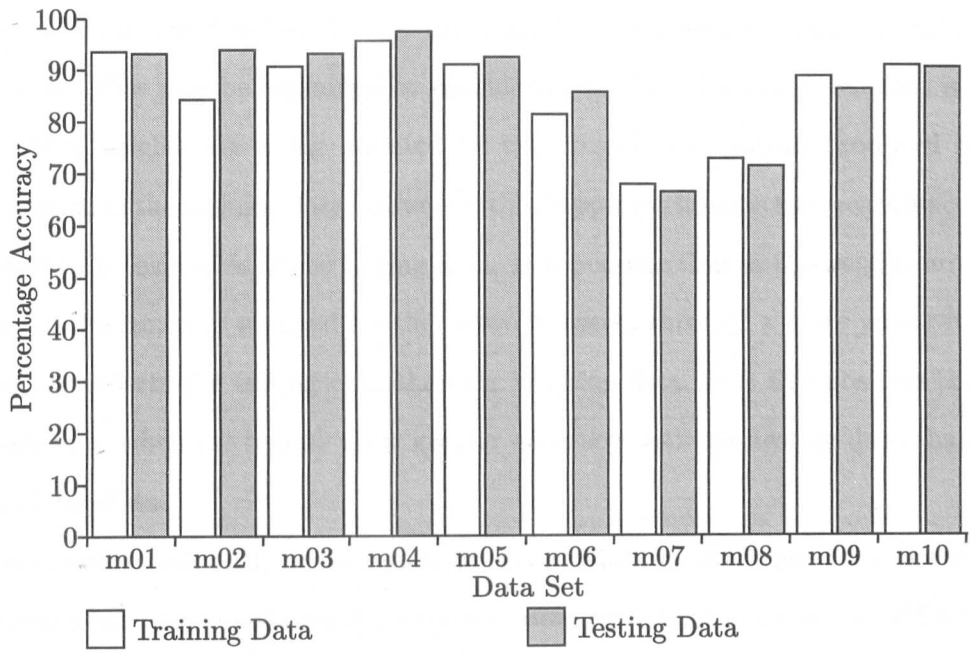


Figure 5.8 Moustache classification - comparison of the different data sets using a constant MLP topology and learning rate

data sets but with the same multilayer perceptron topology and learning rate, in this case networks with 10 hidden neurons and a learning rate of 0.05. These two parameter choices are in the middle of the ranges that were used for each and so serve as "typical" values.

It can be seen that the greyscale data sets performed consistently well, though the best was set m04, the one with the greatest number of elements per training pattern. In addition, of the greyscale data sets, the worst performance is achieved by data set m02, the one with the least elements per training pattern. There are two factors that can relate the accuracy to the number of elements per pattern. Firstly, with more elements, there are more input neurons in the network and therefore more weights which can be used to map the function more accurately. Secondly, in performing the sub-sampling that is done to reduce the image size presented to the network, some information will be lost, therefore the data sets that use less sub-sampling will retain more of the original information.

Once more, the hue data sets show more variation in the results when comparing one data set with another. However, unlike the statistical classifiers, in these results, the

testing and training data sets for any given data set number gave similar classification accuracies. This may be explained by considering the learning algorithm that is being used. The weights are being updated by the error in the output produced by the presentation of the training data. However, the stopping criteria is the greatest accuracy achieved by presentation of the testing data. It is possible that as the weights are being adjusted, the function mapped by the network passes through a state where it more accurately reflects the testing data than the training data. It is this position that the network is in when the results show greater accuracy with the testing data than with the training data.

Due to the consistently worse results produced when using the hue data, the remaining comparisons for moustache classification using multilayer perceptrons will consider only greyscale data.

Graphs presenting the variation in results caused by changing numbers of hidden neurons and altering the learing rate are given in Appendix C. The conclusions resulting from these graphs are given here.

Figure C.1 gives a comparison of the different numbers of hidden neurons that were tried in solving this problem. In this case, data sets m01 and m04 have been selected and the results plotted for all five of the different learning rates that were used. In this graph, the vertical scale has been altered such that it only runs over the range 90–100%. It is evident from the graph that there is little in the way of trends that can be deduced from this study except to say that there is generally a lower accuracy achieved with zero hidden neurons i.e. no hidden layer at all. Having said that, the reduction is only generally around 2% showing that the problem is indeed nearly linear as suggested by the statistical classifiers.

Figure C.2 presents a comparison of the five different learning rates that were used in the network simulations. The graphs show classification accuracies achieved with the m01 and m04 data sets, with the results plotted for all topologies. This shows a peak in the accuracies achieved when the learning rate is 0.1 and a minimum when the learning rate is 0.02. In addition to this, with the higher learning rates, there is a tendency for the accuracy from the testing data to be greater than that from the

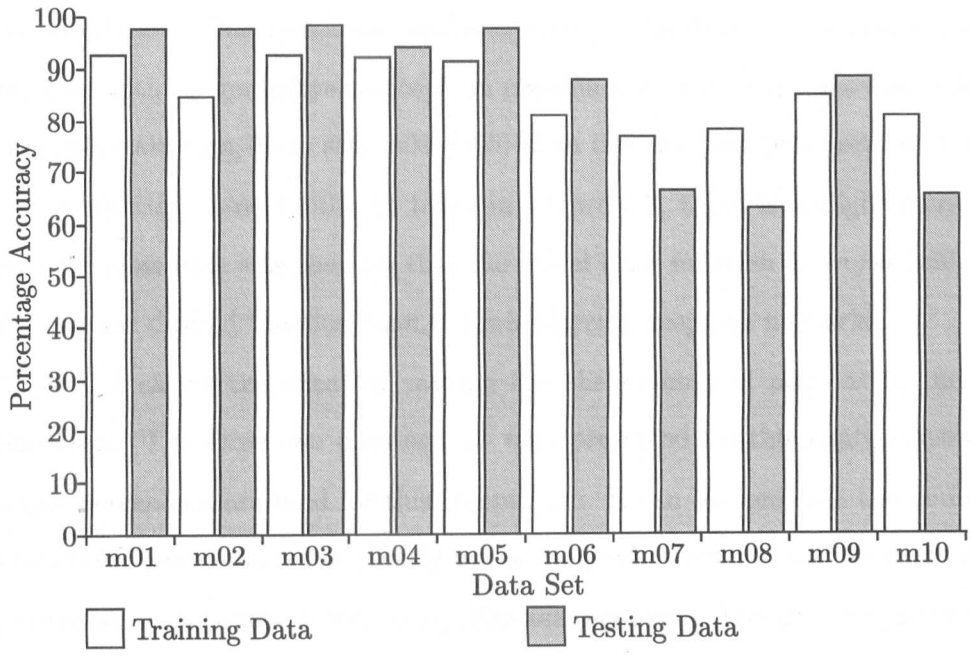


Figure 5.9 Moustache classification - comparison of the different data sets using a constant RBFN topology and learning rate

training data. This may be explained with reference to the comment already made about the learning algorithm and consideration of the gradient descent process used in training (section 3.4.2). With complex problems, the error surface is often not a smooth curve. A large learning rate will cause larger changes in the weights and if this is combined with an uneven error surface, this will cause erratic movement about the surface. This erratic movement will broaden the range of weight combinations that the network passes through during the training process and therefore, if as has been suggested in Section 5.1.2, the testing data is better fitted by a different function to that for the training data, then there is a greater probability of one of these conditions being found. Smaller learning rates will generate more "cautious" movement around the error surface, tending towards the minimum and therefore will not take in such a broad range of weight combinations.

Having dealt with the results from multilayer perceptron simulations, attention is now turned to the radial basis function networks. Figure 5.9 presents a data set comparison using radial basis function networks with 30 neurons in the basis function layer and a learning rate of 0.05. These were middle values of the ones used in the

various simulations thus providing representative results from the complete set. As was the case with the multilayer perceptron classifiers, better accuracies were achieved with the greyscale data (data sets m01–m05) than the hue data (data sets m06–m10). Overall, comparing these results to those in Figure 5.8, there is a slight increase in accuracy for most data sets showing that the radial basis function networks achieved a better fit to the desired function than the multilayer perceptron networks.

Figure C.3 shows the effect of varying the the number of neurons in the basis function layer. The same two data sets as were presented for this analysis using the multilayer perceptron are used for this comparison. It can be seen that the number of basis function neurons does not greatly affect the performance of the network, except for a decrease in accuracy when using only ten neurons. The greatest accuracy is achieved with 40 neurons in the basis function layer and it can also be seen that with this topology, there is less difference in the accuracies achieved with the training and testing data sets than with MPLs.

Figure C.4 shows a comparison of the effect of different learning rates on the output layer for the radial basis function networks. Unlike the multilayer perceptron networks, there is little variation in accuracy due to a change in learning rate though the values 0.05, 0.1 and 0.2 do perform better than the other two. Overall, the best results are probably found using 0.1 as the learning rate for the output layer. The radial basis function network effectively re-maps the input data into a different space by use of the basis function layer. The un-supervised learning that is used on this layer uses clustering statistics to assign the basis centres and since it has been observed using PCA that the data can cluster according to the desired classification, it may be expected that the results from a radial basis function network will be good. As the basis function layer is already reflecting the structure of the data, the output layer of these networks has a simpler function to represent than is the case with the multilayer perceptron. This explains why the results are not significantly affected by the learning rate for this layer. With a simpler function, the error surface is likely to be smoother so the gradient descent method of training will operate more as expected.

Overall, the radial basis function networks achieve slightly greater accuracy than the multilayer perceptron networks when trained to recognise the presence of moustaches. Which of these two network architectures performs better is dependent on the function that the network is being trained to simulate. The radial basis function network build functions by the superposition of a collection of basis functions whereas the multilayer perceptron uses a collection of sigmoidal functions. Therefore the relative performance of the two will depend on how easily the data can be modelled by a combination of these particular functions.

5.2.2 Beard classification

The same statistical and neural techniques were applied to the problem of beard identification and the results from these experiments are presented in the following sections.

Statistical classifiers

Since this problem, like that of moustache identification, is a multiple input, single output problem, the linear regression technique can also be applied to the beard data sets.

Following the same technique as before, the results are shown in Figure 5.10. As with the moustache identification, these results are presented with the data sets grouped such that the first five use greyscale data while the second five use hue data. As before, it can be seen that greater accuracy is achieved with the greyscale data than with the hue. It can also be seen that within the two groups, the highest accuracies are achieved with data sets b02 and b01. These are the two sets with the greatest number of elements per vector, showing a similar effect to that noted with the moustache data when using linear regression, that is, a large number of parameters in the regression model giving a closer fit to the data.

The accuracies for the Euclidean distance classifier are given in Figure 5.11. Once more, a greater accuracy is observed with the greyscale data than with the hue data, though in this instance it is not so significant. Within the two groups, the accuracies are very similar, particularly for the training data with the greyscale images. It is

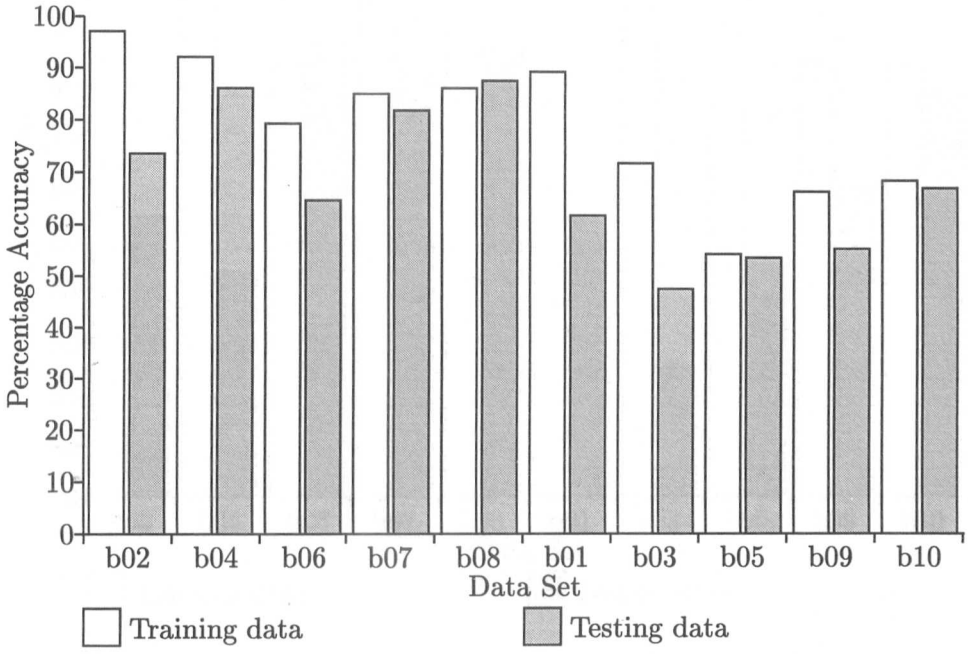


Figure 5.10 Linear regression classification results for beard data sets

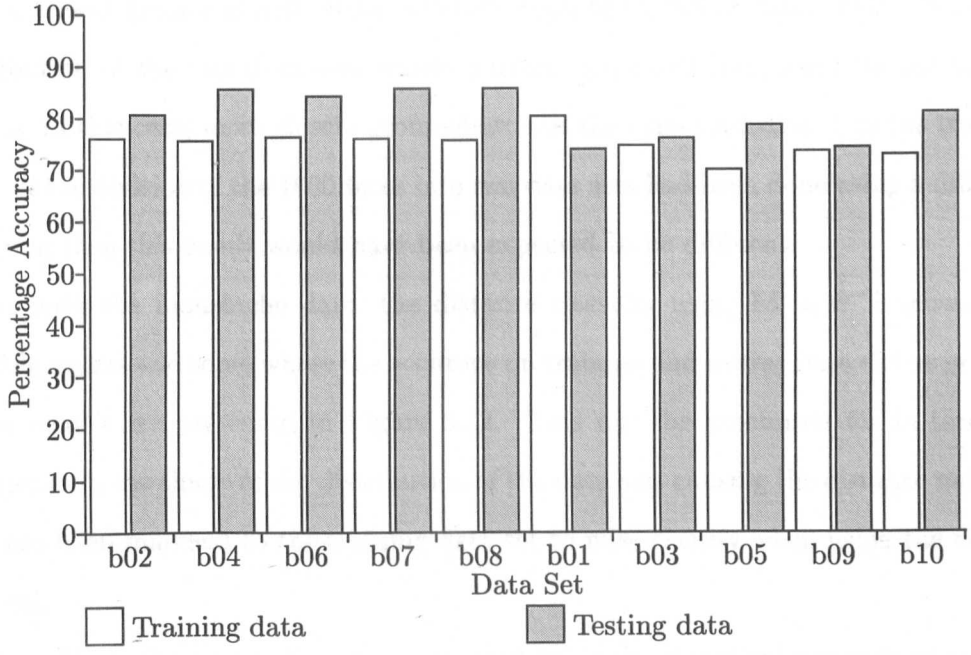


Figure 5.11 Euclidean distance classification results for beard data sets

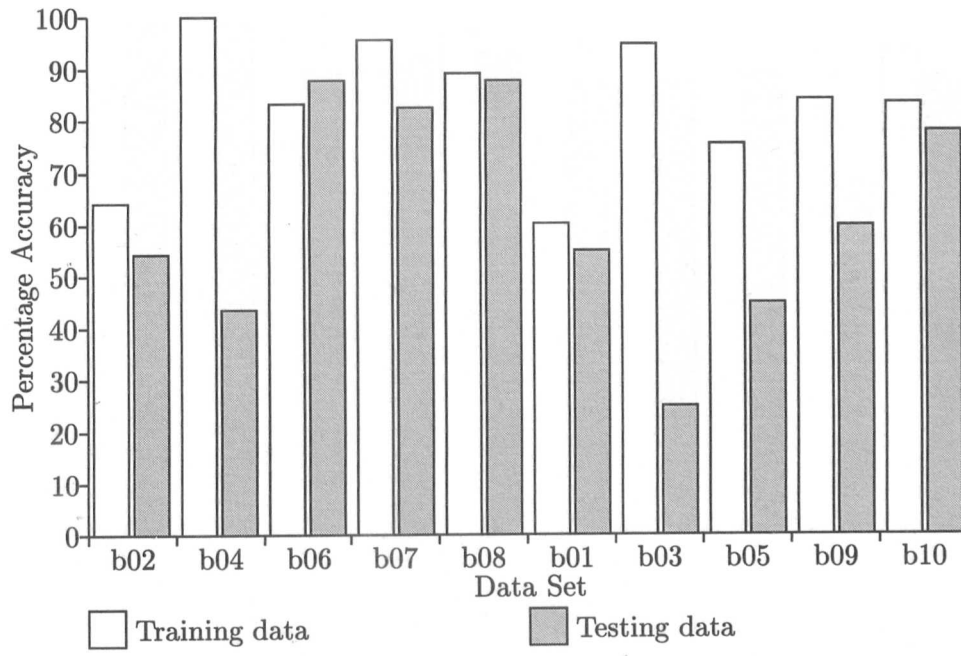


Figure 5.12 $\mathbf{M} = \mathbf{S}_i^{-1}$ distance classification results for beard data sets

interesting to note that with the exception of b01, in each of the cases here, the testing data achieved greater classification accuracy than the training data. This reflects the distribution of the two data sets within pattern space and indicates that the testing data is, in this case, more closely grouped around the class centroids than the training data. If the division of the 1000 faces into two data sets had been done using a different grouping then this result would have been expected to be different.

As with the moustache data, the distance classifier using $\mathbf{M} = \mathbf{S}^-1$ shows very variable results and some where the accuracy on training and testing data differs greatly. These results are presented in Figure 5.12. This may be accounted for in terms of differences in the shape of the distribution of the data sets causing the distance measure that has been matched to the training data set to miss-classify when using the testing data set.

Overall, similar results have been achieved using the statistical methods when considering the beard data sets as were achieved with the moustache data sets. This finding is also supported by the principal components analysis plots (see Section 4.4.1), which,

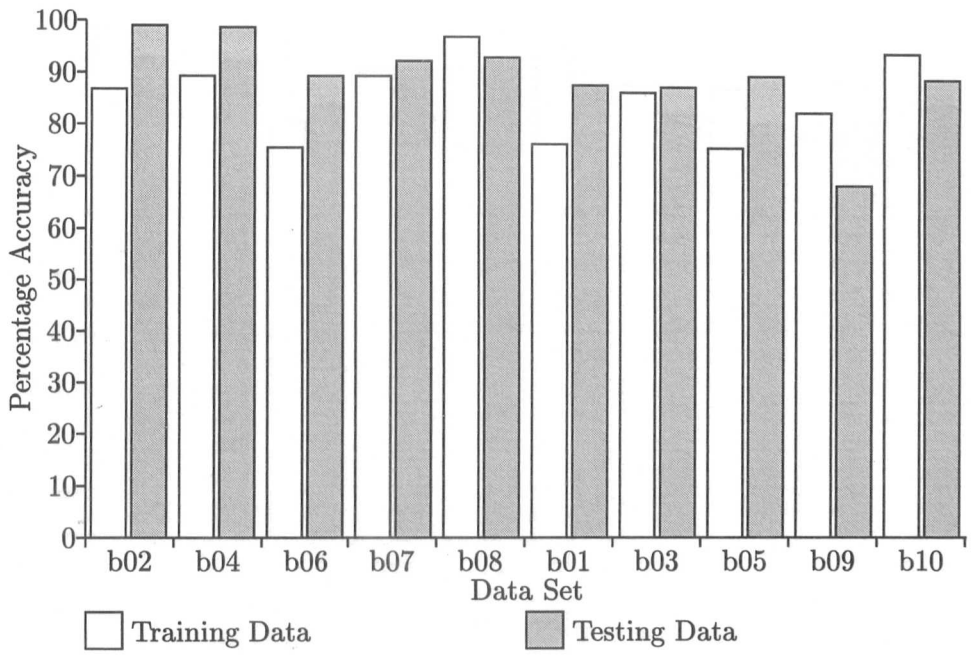


Figure 5.13 Beard classification - comparison of the different data sets using a constant MLP topology and learning rate

although not as clear as in the moustache identification case, do show some separability of the data into the two classes.

Neural network classifiers

Turning to the use of neural network classifiers for beard identification, Figure 5.13 shows a comparison of the performance of the different data sets using a multilayer perceptron with ten hidden neurons and a learning rate of 0.05, the same values as were used for this comparison when using the moustache data sets. The accuracies achieved here are again high, around the 90% mark though with greater values obtained with the greyscale data than with the hue data. Looking at the results for the greyscale data sets (b02, b04, b06, b07, b08) it can be seen that there is some correlation between the number of elements per vector and the result achieved in that the first two data sets are those with the greatest number of elements and they give the best results. This relationship is not carried through to the hue data sets where the best result is from data set b10, one with 20 elements per vector, the smallest number used in these experiments.

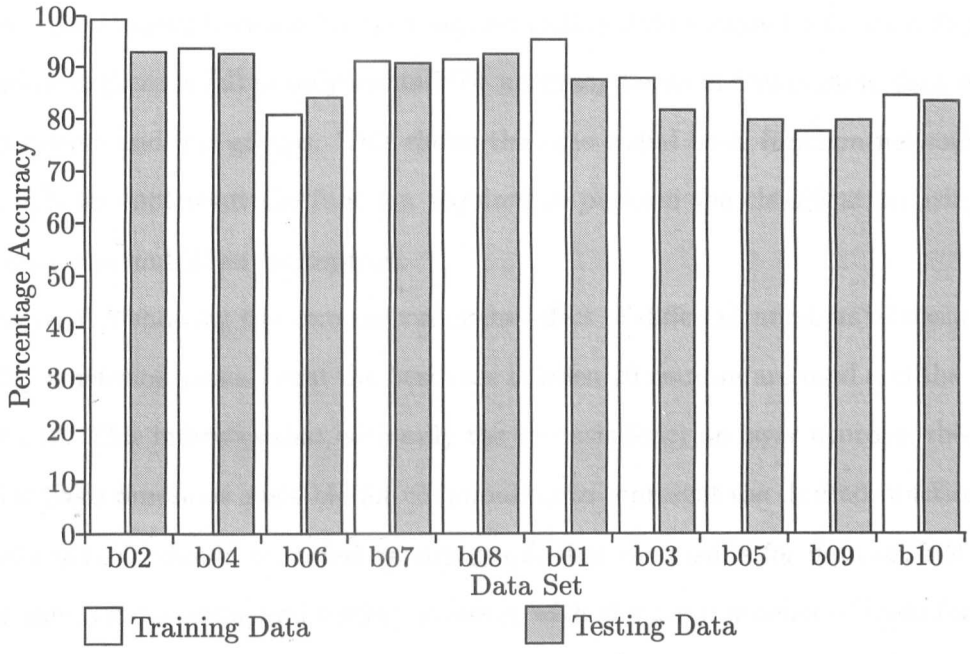


Figure 5.14 Beard classification - comparison of the different data sets using a constant RBFN topology and learning rate

For the remaining examination of multilayer perceptron classifiers, attention will only be given to the greyscale data sets since these have proved to give greater accuracy than the hue data sets. Figure C.5 shows the effect of varying the number of hidden neurons. Two data sets are presented each with all five learning rates. The graphs show little variation in the accuracies achieved as the number of hidden layer neurons is varied and once more the effect of the training algorithm's stopping criteria can be seen in that the testing data regularly gives greater accuracy than the training data.

Figure C.6 presents the results of varying the learning rate for each of the multilayer perceptron topologies for the same two data sets. A similar effect is observed here as was seen for the moustache data sets with a fall in the accuracies at a learning rate of 0.02 and better results achieved with higher learning rates. It can also be seen that with the b02 data set, the learning rate of 0.02 results in a much closer relationship between the accuracies from the training and testing data sets.

The last combination to be examined is the radial basis function network for beard identification. Figure 5.14 gives the data set comparison when using a network with 30 basis function layer neurons and a learning rate of 0.05 on the output layer. This

shows a good match between the training and testing data accuracies for each data set. In addition, there is fall of only around 5% accuracy between comparable data sets in the greyscale and hue groups. This shows that the radial basis function networks are more able to implement the function required to perform the classification using hue data than the multilayer perceptron.

Figure C.7 showing the comparison of the effect of different numbers of basis function layer neurons, reveals that the best case is when 20 neurons are used and the worst is with 10. This indicates that with only the 10 basis function layer neurons, there are too few basis functions available for combination to represent the desired function sufficiently well. It should be noted on this graph that the results for data set b04 show closer values for training and testing accuracy with any given number of basis function neurons than is the case with data set b02. This may be explained by the fact that data set b02 has 100 elements per vector whereas data set b04 has 50. The difference between the two is that data set b02 is taken over a wider section of the original image and will take in areas outside the chin. These could cause the training and testing data sets to differ in terms of the function that they represent as these areas would not necessarily be assumed to be relevant to the classification of beards.

Finally, Figure C.8 shows the effect of varying the learning rate for the output layer of neurons. The familiar pattern of a minimum at 0.02 is seen once again with 0.05 giving the best overall results for these networks.

5.3 Conclusions

The three statistical classifiers that were considered each showed that they could be used for the task in hand though none performed as well as the neural networks. The linear regression method revealed that for both the moustache and beard classifications, the greyscale data was considerably more linear than the hue data and this pattern was repeated with the neural network classifiers. The simple Euclidean distance classifier performed well on all data sets with accuracies around 80% while the distance classifier using the inverse covariance matrix was very variable in its performance. This suggests that the data sets here are not well represented by the model used in this form of

classifier.

There are an almost limitless number of parameter combinations that can be set for the neural network simulations discussed here. All that can realistically be achieved is an evaluation of a limited number of different combinations which are conventionally found to be appropriate to good ANN design. The simulations performed here have shown that both multilayer perceptrons and radial basis function networks may be used in the identification of moustaches and beards.

In varying parameters associated with the neural network models, little could be established in terms of ideal values. The number of hidden neurons or basis function neurons only made a noticeable change to the accuracy when it was reduced to a particularly small number (or zero). Learning rate was generally best when set to 0.05 or 0.1. These comments are unfortunately typical of the results that are found when using neural networks - there is often no "best" set of parameters and many choices will perform sufficiently well.

To summarise, these experiments have shown the practical possibility of using neural networks to identify the presence or absence of moustaches and beards from sections of greyscale images of faces.

Chapter 6

More Complex Feature Classification

The majority of the feature measures in the Aberdeen data can take more than two values. The simple cases of moustache and beard identification presented in Chapter 5 are the exception in having bi-valued results, but have been used in this thesis as an exploration of possible techniques for evaluating facial features. The feature that will now be examined is eye colour. This one has a total of five possible values in the descriptive measures defined by Aberdeen University. It is in fact unique in this respect as all the other multi-valued measures take values on a continuous scale from 1 to 5 to an accuracy of one decimal place. A value of 0 is used in cases where the feature cannot be measured, for example if the feature is covered by hair.

This particular feature was chosen from the list of features available as while it is one of the more "difficult" features to classify, it is clear what area of the image needs to be examinied to perform the classification and the eyes are often listed as one of most used features in the human recognition process. It also brings into the work the use of colour which is not immediately relevant to some of the other non-physically derived measures.

Eye colour is not simple to classify. Just a quick examination of a few peoples' eyes will show that the colours are not uniform within the eye and some examples are difficult to categorise into one of the five classes that is used here. Also, as there are no

"hard" boundaries between the different eye colour classes, there is no guarantee that the original classifications are the ones that another set of jurors would choose. For these reasons 100% accuracy can never be expected for this problem and in fact such high accuracy may not be desirable in a problem that is not well defined such as this one.

With the problem of eye colour, advantage may be taken of the fact that there are two eyes on each face and therefore twice as many examples as were present for the moustache and beard cases. Training data has been taken from 900 right eye images and testing data from the remaining 100. In an attempt to test the classification systems in as true to life manner as possible the 1000 left eye images have been used to produce a third data set for validation (see Section 3.5.1). While using the other eye to generate a validation data set may not be ideal, due to the inevitable link between the colours of two eyes belonging to the same person, it was concluded from examination of the images that there is sufficient variation to warrant this approach. There is a degree of natural variation between the colours of a person's left and right eyes and additional variations are introduced by the lighting conditions used.

6.1 Experimental Method

The data sets used for this work have been analysed with both PCA and Kohonen SOM networks as presented in Chapter 4. This has shown that the problem of eye colour classification is a complex one. Similar methods to those used for the moustache and beard classification are applied to this classification problem though changes have to be made to accommodate the additional number of possible output values.

6.1.1 Statistical Classifiers

For the moustache and beard classification problems, two statistical methods were employed as classifiers to use as a benchmark against which the neural methods could be evaluated. In this five class case, the linear regression method is no longer appropriate due to the output representation needed. Output for a five class system can be represented in two main ways. One method is to have a system with one output variable

to identify each class, and the system is designed such that only one of these variables gives a "true" output while the others give a "false" output. For the linear regression case, this would result in the need to find five functions, which is perfectly possible. The problem that arises is in the distribution of the output values. If the data sets are balanced with equal numbers of examples of each class then, for the five class case, each output will be "false" four times as often as it is "true". In a simple linear regression system, this will yield a function which will return an almost constant "false" value regardless of the input since that is the value most frequent in the training data.

An alternative output representation is where a single output is used, with different values for each of the classes. This is possible if there is some clear relationship between the classes such that "like" classes can be arranged to have similar output values. In other cases where there is no obvious relationship between the classes, it is not clear as to how to order the classes when deciding on the output values for each class. For example in the case of eye colour, should the coding of a single output variable be 1 - blue, 2 - grey, 3 - green, 4 - hazel, 5 - brown as in the data from Aberdeen or should some other arrangement be used? For the purpose of providing a comparison, we will assume that this is the best order of the classes; this was, after all, the order devised for searching the descriptive measures index and therefore has been devised with "similar" colours represented by adjacent numbers.

The distance classifiers on the other hand are not affected by output representation. They simply identify which of a number of class centres any given vector is closest to.

6.1.2 Neural Network Classifiers

A natural extension of the neural methods used for the moustache and beard classification is to produce networks with five outputs, one per eye colour class. Other arrangements are possible such as a single output with a graded response or three outputs, whose combination is used to determine the classification, however these all imply a greater relationship between the colour classes than is the case when five outputs are used. The use of five outputs means that mathematically, all the classes are treated equally within the neural network with noparticular association with each other caused

by the network topology. It has already been established in Chapter 4 that the eye colour problem is complex and so it will require more free parameters in the neural model in order to solve it, that is, more hidden neurons and thus more weights will be needed than was the case for the moustache and beard data sets. For each of the data sets, networks with 5, 8, 10, 15, 20, 25, 30, 35, 40 and 50 were used and as with the previous experiments, each was run five times with each of six different learning rates. This combination resulted in a total of 7500 simulations. For the radial basis function networks, networks with 25, 50, 75, 100 and 150 basis layer neurons were used, each simulated five times with each of five different learning rates being used on the output layer giving a total of 3125 simulations.

With a five output network, the analysis of the results is more complex than with the single output networks used previously. Decisions need to be made as to how the five outputs are to be interpreted to give a single colour as the system output. Algorithm 6.1 presents the algorithm that was used. It is based on each line of the output file containing ten entries; the first five being the target outputs and the rest being the real outputs.

A more complex version of this algorithm was also used for the production of confusion matrices. Rather than incrementing one of three counters (*true*, *false* and *undetermined*), the entry in a matrix corresponding to the appropriate target and actual output class was increased after the identification of these two classes.

All the analysis algorithms were performed either using scripts writen by the author in Perl or Matlab routines. The NeuralWorks simulations of the networks produced output text files containing the network output values and the desired output values for the and these were supplied as the input to the analysis routines. Perl scripts were used for most of the text file manipulation work as this is one of the strengths of the Perl language. Matlab programs were added to the system to facilitate the plotting of some of the graphs, primarily the line graphs. The bar charts were produced using a further Perl routine.

Algorithm 6.1 Interpretation of five output networks

Require: `Entries in each line of the output file are` *elem*`[1],` *elem*`[2], ..., elem[10]`

true ← 0 {Number of correct results}
false ← 0 {Number of wrong results}
undetermined ← 0 {Number of undetermined results}
threshold ← 0.2

for each line in the output file **do**
  target ← 0
  {Identify the target class}
  for $i = 1$ to 5 **do**
    if *elem*[*i*] == 1 **then**
      target ← *i*
    end if
  end for

  actual ←
  {Identify the actual class given by the network}
  next_biggest ← -1 {*next_biggest* will hold the second largest output value}
  for $i = 6$ to 10 **do**
    if *elem*[*i*] ¿ *elem*[*actual*] **then**
      actual ← *i*
    else
      if *elem*[*i*] ¿ *next_biggest* **then**
        next_biggest ← *elem*[*i*]
      end if
    end if
  end for

  {If the largest output is at least *threshold* bigger than the next largest then check for correct match - otherwise result is undefined.}
  if *elem*[*actual*] > *next_biggest* + *threshold* **then**
    if *target* == (*actual* - 5) **then**
      true ← *true* + 1
    else
      false ← *false* + 1
    end if
  else
    undetermined ← *undetermined* + 1
  end if
end for

accuracy ← (*true* × 100)/(*true* + *false* + *undetermined*)

6.2 Results

Throughout the results section, data sets have been grouped according to type. Examination of Table 4.7 will reveal that the numerical name of each of the eye*nn* data sets is not related to a particular combination of image area and colour representation. For this reason, when results are presented comparing the performance of the different data sets, the sets will not appear in numerical order. While in the two class case of moustache and beard classification, a random number based classifier would have achieved 50%, in this five class case, the figure is 20%.

6.2.1 Statistical Classifiers

Using the encoding method defined in the Aberdeen database, a simple linear regression model was implemented in Matlab for each of the 25 eye colour data sets using a similar routine to that used in Section 5.1.1. Figure 6.1 shows the results achieved with this model. The data sets are grouped by colour representation, in the order "grey", "hue", "red", "green" and "blue". Within each colour group, the data sets are ordered according to the number data elements in the input vector; the largest coming first.

To provide a means of comparison with the neural network results presented later, the "validation" accuracy has been included with this analysis. In the two class cases presented in Chapter 5, a threshold of 0.2 was used to determine which classifications were undetermined. For that work, the output was specified to be in the range $-1, 1$, that is the desired outputs were -1 and 1. In this five class case, the desired outputs are 1, 2, 3, 4 and 5 hence the numerical difference between two adjacent classes is half that which it was in the former case. Therefore for this five class problem, the threshold

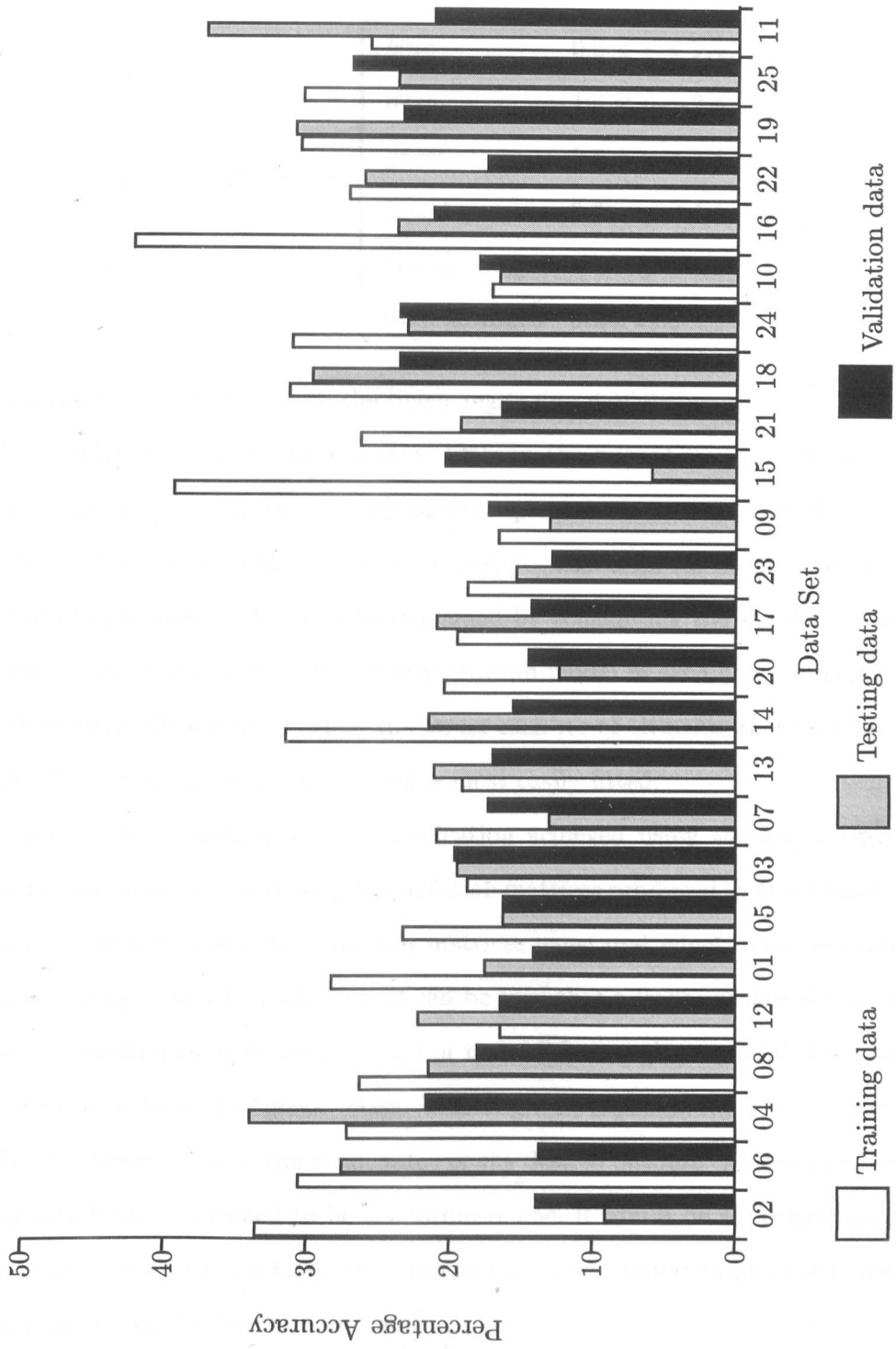


Figure 6.1 Linear regression classification results for eye colour data sets

has been set to 0.1 giving the classification to be defined as

$$output\ class = \begin{cases} blue & 0.6 < y < 1.4 \\ grey & 1.6 < y < 2.4 \\ green & 2.6 < y < 3.4 \\ hazel & 3.6 < y < 4.4 \\ brown & 4.6 < y < 5.6 \\ \text{undetermined} & \text{otherwise} \end{cases} \tag{6.1}$$

where y is the output value from the linear regression model.

The results given in Figure 6.1 show little in the way of trends. One exception is that it can be seen that within each colour representation group, it is the data sets with the largest number of elements per input vector that achieves the greatest accuracy with the training data. This may be explained by considering the number of degrees of freedom that is involved in the linear regression model in each case. With the data sets which have 140 input elements, the larger number of elements there are the most degrees of freedom and hence the model is most easily fitted.

A better understanding of the classification achieved using the linear regression method is obtained by considering the confusion matrices produced by this classification process. Table 6.1 shows the confusion matrices generated when linear regression is performed using data set eye02. Here it can be seen that a large number of the vectors are being classified as undetermined and of those that are given a valid classification, many are said to be in the "green" class. This might be expected since the output value of 3 for the "green" class is the mean value of the desired outputs. As the classification has already been established to be a non-linear one, it would be expected that the a linear function may yield a high occurance of the mean desired value as it is not able to adequately map the function in question.

Figure 6.2 shows the results obtained using the Euclidean distance classifier. These reveal little difference between the data sets in terms of classification accuracy using Euclidean distance. The confusion matrices for data set eye02 are given in Table 6.2. In this instance, many of the eyes have been misclassified as brown, indicating that

Table 6.1 Confusion matrices for linear regression classifier using the eye02 data set

Training Data

undetermined	117	110	98	64	118
brown	5	0	0	16	**96**
hazel	38	0	14	**160**	228
green	111	165	**308**	192	72
grey	128	**165**	70	16	10
blue	**66**	0	0	0	2
	blue	grey	green	hazel	brown

Testing Data

undetermined	16	18	10	30	9
brown	0	0	0	15	**9**
hazel	4	0	0	**0**	30
green	15	27	**0**	15	12
grey	12	**9**	40	0	6
blue	**10**	9	10	0	0
	blue	grey	green	hazel	brown

Validation Data

undetermined	165	121	130	207	168
brown	19	22	26	9	**76**
hazel	52	55	78	**72**	128
green	102	99	**52**	126	110
grey	106	**99**	156	90	64
blue	**78**	121	91	36	24
	blue	grey	green	hazel	brown

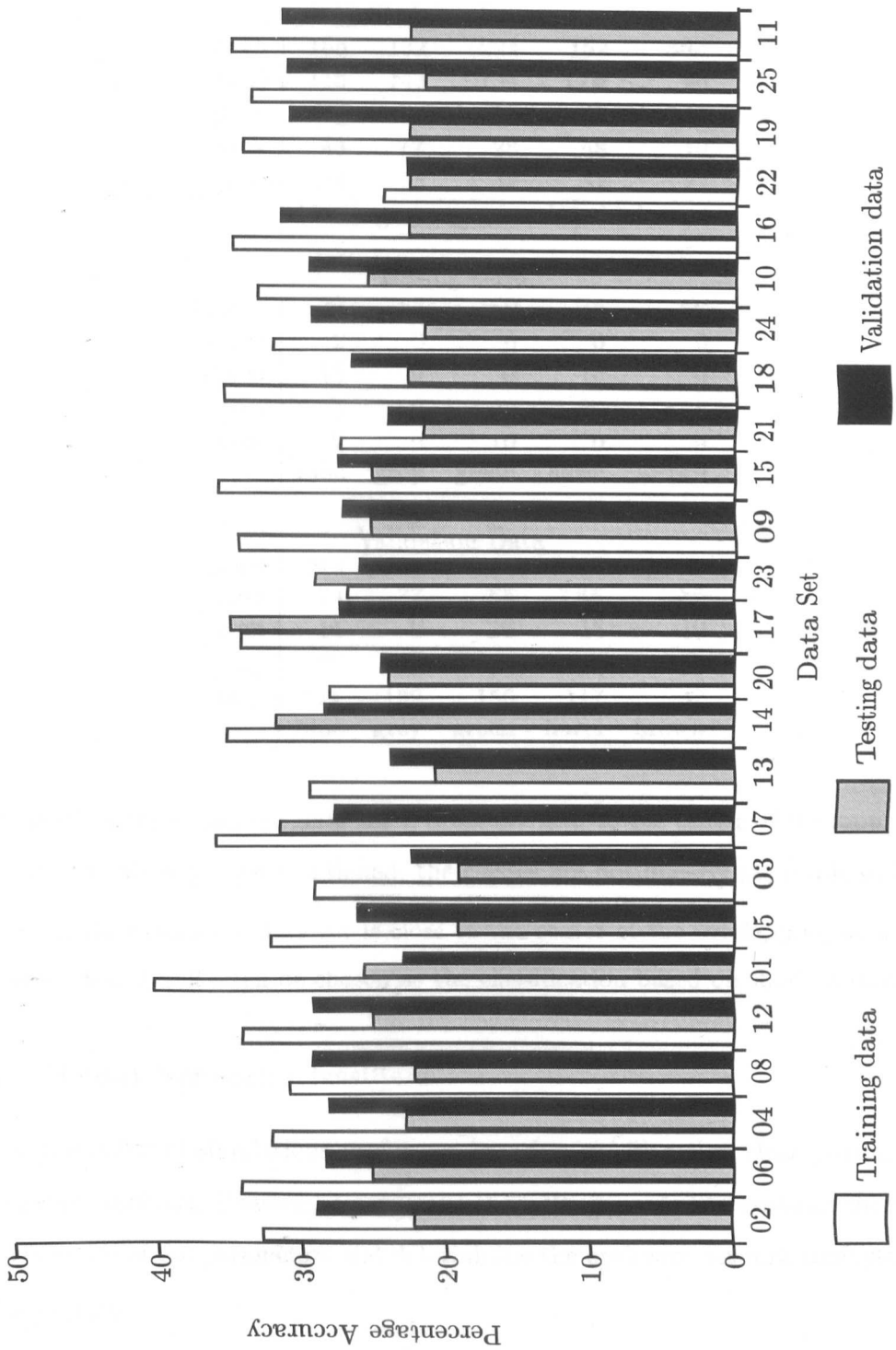


Figure 6.2 Euclidean distance classification results for eye colour data sets

Table 6.2 Confusion matrices for Euclidean distance classifier using the eye02 data set

Training Data

brown	165	132	224	152	**384**
hazel	115	143	154	**176**	96
green	44	33	**42**	40	14
grey	43	**77**	28	48	10
blue	**98**	55	42	32	22
	blue	grey	green	hazel	brown

Testing Data

brown	37	63	50	45	**66**
hazel	2	0	0	**0**	0
green	15	0	**0**	15	0
grey	1	**0**	0	0	0
blue	**2**	0	10	0	0
	blue	grey	green	hazel	brown

Validation Data

brown	219	242	195	216	**412**
hazel	74	77	65	**135**	86
green	11	0	**26**	18	10
grey	77	**66**	91	54	20
blue	**141**	132	156	117	42
	blue	grey	green	hazel	brown

the "center" of the brown eye data set is probably near to the center of the input data space. As has already been mentioned, the classes are not linearly separable so if the "center" of the brown eye data set is close to the center of the overlapping data from all classes, then it will often be chosen as the classification based on shortest distance.

6.2.2 Neural Network Classifiers

The large number of simulations performed here (see 6.1.2) makes clear presentation of the results difficult. However, comparison of results is needed to evaluate the effect of varying each of the parameters and determining the optimum network configuration for this problem.

Taking one network configuration from the ten different hidden layer sizes and six different learning rates as a common standard makes evaluation of the different data sets possible. Figure 6.3 shows results for a series of networks with 20 hidden layer neurons and using a learning rate of 0.05. The best test of these networks is consideration

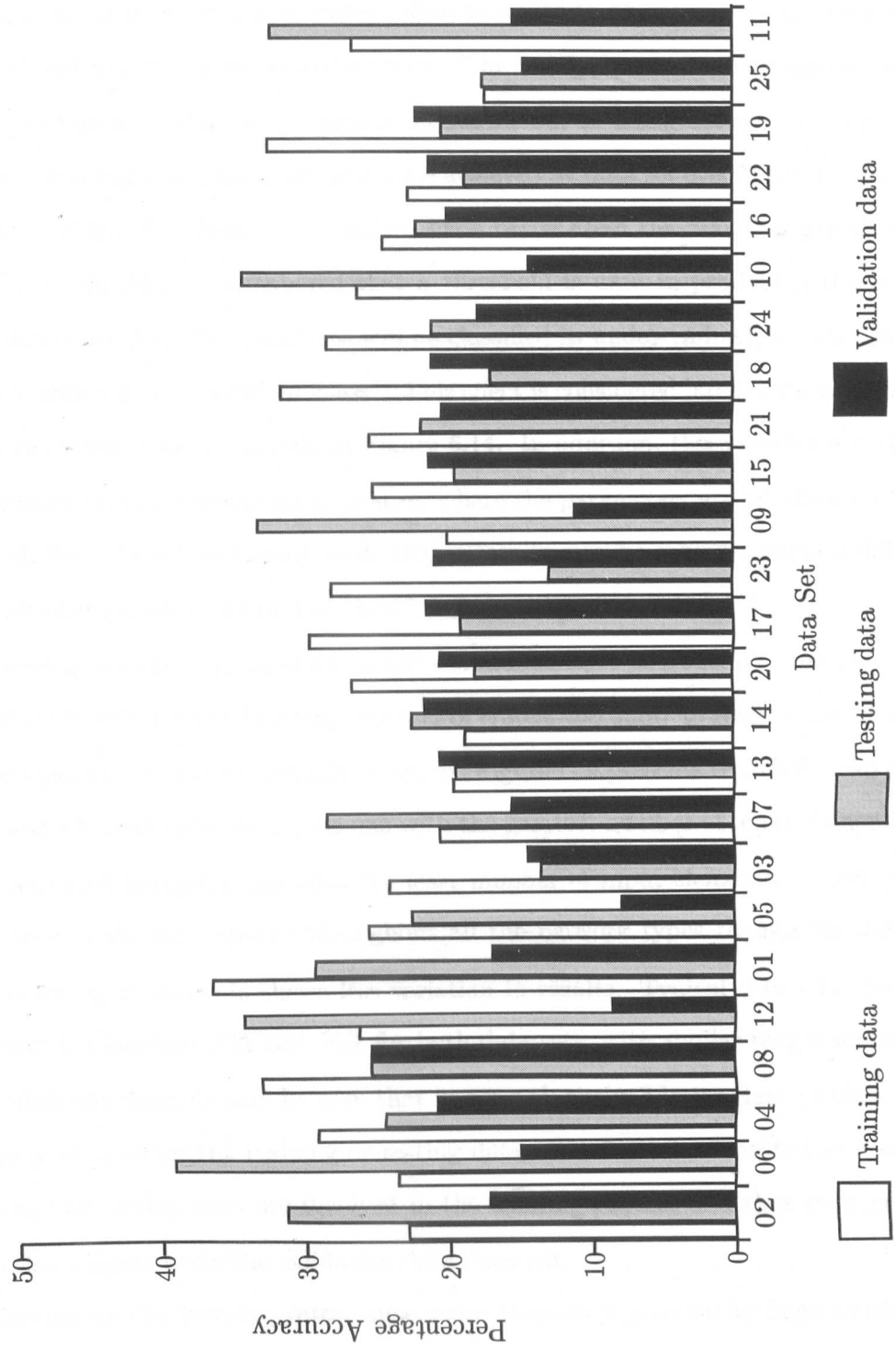


Figure 6.3 Eye colour data set comparison using an MLP

of the accuracy using the validation data as this is the most representative of the network in use with previously unseen data as would be the case were the network to be deployed in a real working environment. Examining the results as groups according to colour representation, no particular set stands out as being better than any other. On an individual basis, the best results are achieved by data set eye08 with a validation accuracy of 25.5%. While this is only a little better than the "random guess" result of 20%, it should be remembered that a threshold is used in producing this results such that a number of the patterns will be classified as undetermined, a situation that doesn't arise with the "random guess". This effect is considered later in the comparison of all the classification methods in Figure 6.14. In addition, these results are simply taken from a single network configuration where the parameters are middle ones taken from all the different variations used. Better results may be achieved using a different network configuration. Again the "best" results are presented later.

Turning to a comparison of the number of hidden layer neurons, plotting graphs for all data sets would result in a huge number of traces, too many to realistically compare so once more a subset of the results is given. Figure C.9 presents results for two of the data sets; data set eye02 being the one with the greatest number of input elements and data set eye09 being the one with the least number of input elements. These graphs both show a similar response throughout all the network types though the data set with more input elements shows less variation in results. Typical values for training accuracy fall between 20% and 30% for both data sets, with similar ranges for testing and validation data. It may be seen that in general, the validation data yields a lower accuracy than either the training or testing data. This is to be expected as both the training and testing data are involved in the training process and thus influence the network's weights while the validation data does not.

Comparing the learning rates, once more there is a potentially huge number of graphs that could be plotted. Figure C.10 shows graphs for the same two data sets as used for the hidden layer size comparison. The first, data set eye02, suggests that best results were achieved with the lower learning rates of 0.01, 0.02 and 0.05 with overall accuracies falling by around 5% for the higher learning rates. With data set eye09, the

results are spread over a wider range as the learning rate becomes larger. This may be explained by considering the learning process as a search for the global minimum of the error surface.

The learning rate determines the size of steps that is taken around the surface in searching for the minimum. If the problem in question has a "rough" error surface, that is one that does not follow smooth curves, then large steps around the error surface as produced by a large learning rate, will not necessarily follow a path that descends the error surface. Rather, the training process will jump around the error surface in such a manner that the error after any given training step, is not related to the error value prior to the training step.

Effectively what is produced is a form of almost random search where much is dependent upon the initial conditions of the network weights. Under these conditions, there will be a much greater distribution of the results as some network training runs will randomly achieve better results than others.

In order to give a clearer picture of the classifications that are being performed, Table 6.3 shows the confusion matrices generated by network which gives the greatest validation accuracy using data set eye02. These matrices show that despite the balancing of the data sets to include equal numbers of examples of each class, the classification process has developed a bias towards the brown and hazel classes and the green class is rarely chosen. This is a similar result to that obtained using the Euclidean distance classifier as presented in Table 6.2 and unlike the linear regression case given in Table 6.1 where it was the green class that was selected the most frequently. This indicates that the multilayer perceptron network is governed by the distribution of the data points within the input space in a similar manner to the Euclidean distance method, rather than being affected by the output coding as was the case with the linear regression.

With the different colour representation being used as input data, it is possible to examine which of these best identifies each of the eye colour classes. Table 6.4 shows one confusion matrix for each of the colour representations. These are taken in each case from the network that gave the best accuracy with validation data using the given colour representation. The figures in brackets are the validation percentage accuracies

Table 6.3 Confusion matrices for MLP network classifier using the eye02 data set

Training Data

undetermined	29	44	70	0	36
brown	64	44	98	72	**400**
hazel	150	88	140	**368**	58
green	8	0	**154**	0	6
grey	39	**242**	14	8	18
blue	**175**	22	14	0	8
	blue	grey	green	hazel	brown

Testing Data

undetermined	6	18	0	0	0
brown	9	0	0	0	**48**
hazel	14	18	20	**45**	9
green	6	0	**10**	15	0
grey	3	**9**	0	0	3
blue	**19**	18	30	0	6
	blue	grey	green	hazel	brown

Validation Data

undetermined	49	66	65	27	36
brown	108	121	156	153	**410**
hazel	122	154	104	**189**	74
green	5	0	**13**	18	8
grey	103	**99**	65	90	18
blue	**135**	77	130	63	24
	blue	grey	green	hazel	brown

Table 6.4 Validation data confusion matrices for eye colour classification using an MLP network and different colour representation data sets

Hue Data (25.9%)

undetermined	55	66	104	9	48
brown	100	77	52	63	**232**
hazel	269	319	273	**432**	192
green	37	11	**52**	9	38
grey	8	**33**	26	0	14
blue	**53**	11	26	27	46
	blue	grey	green	hazel	brown

Grey Data (28.4%)

undetermined	29	33	26	9	16
brown	96	154	104	171	**398**
hazel	279	231	312	**342**	134
green	22	11	**13**	9	10
grey	10	**0**	13	0	0
blue	**86**	88	65	9	12
	blue	grey	green	hazel	brown

Red Data (24.4%)

undetermined	56	55	78	72	76
brown	142	209	156	126	**300**
hazel	134	154	104	**198**	92
green	57	33	**78**	18	36
grey	122	**55**	117	117	58
blue	**11**	11	0	9	8
	blue	grey	green	hazel	brown

Green Data (30.4%)

undetermined	18	11	13	18	6
brown	39	44	26	45	**324**
hazel	289	385	312	**414**	222
green	0	0	**0**	0	0
grey	92	**66**	117	45	8
blue	**84**	11	65	18	10
	blue	grey	green	hazel	brown

Blue Data (27.1%)

undetermined	16	33	65	54	22
brown	98	121	208	306	**494**
hazel	12	22	26	**45**	22
green	0	0	**0**	0	0
grey	1	**0**	0	0	4
blue	**395**	341	234	135	28
	blue	grey	green	hazel	brown

for each network.

From these matrices, it can be seen that no particular colour representation stands out as being more suited to the purpose than any of the others. However, we can see here which colour representations are particularly suited to identifying given eye colour. The best results for each eye colour class are marked by a box around the number. In all cases, the green and grey eyes fair badly, they appear to be the most difficult to identify. In addition, several of the networks appear to be biased towards one class for example, the "hue" data set identifies more eyes as hazel than as any other colour. This indicates a basic problem with the learning procedure whereby a single target is being learnt more easily than others.

Having examined the results achieved using multi layer perceptron networks, consideration is now given to the performance of radial basis function networks using the same data sets. In Figure 6.4 the comparison of the results achieved with different data sets using a network with 75 basis function neurons and a learning rate of 0.05 on the output layer is shown. As previously, the data sets are grouped according to colour representation and by input vector size within each group. This graph does not show any clear trends in terms of which colours or vector size produces the best results, though it can be seen that the worst results are achieved by two data sets using the "hue" colour representation. Particularly good results are achieved by data sets eye01, eye15 and eye24. Comparing these results with those presented in Figure 6.3 where a multi layer perceptron was used, there are ten cases where the RBFN out performs the MLP and therefore fifteen where the MLP out performs the RBFN.

It must be remembered that this comparison involves two networks of different types where there is no established means of performing a direct comparison, that is for a given multilayer perceptron, there is not a particular radial basis function network that may be considered to be equivalent. A alternative comparison is given in Figure 6.5 where the best validation result for each data set is given for the two network types. In this comparison it can be seen that generally the two network types perform similarly though for seventeen of the data sets, the multi layer perceptron performs better than the radial basis function network. Thus the multi layer perceptron has

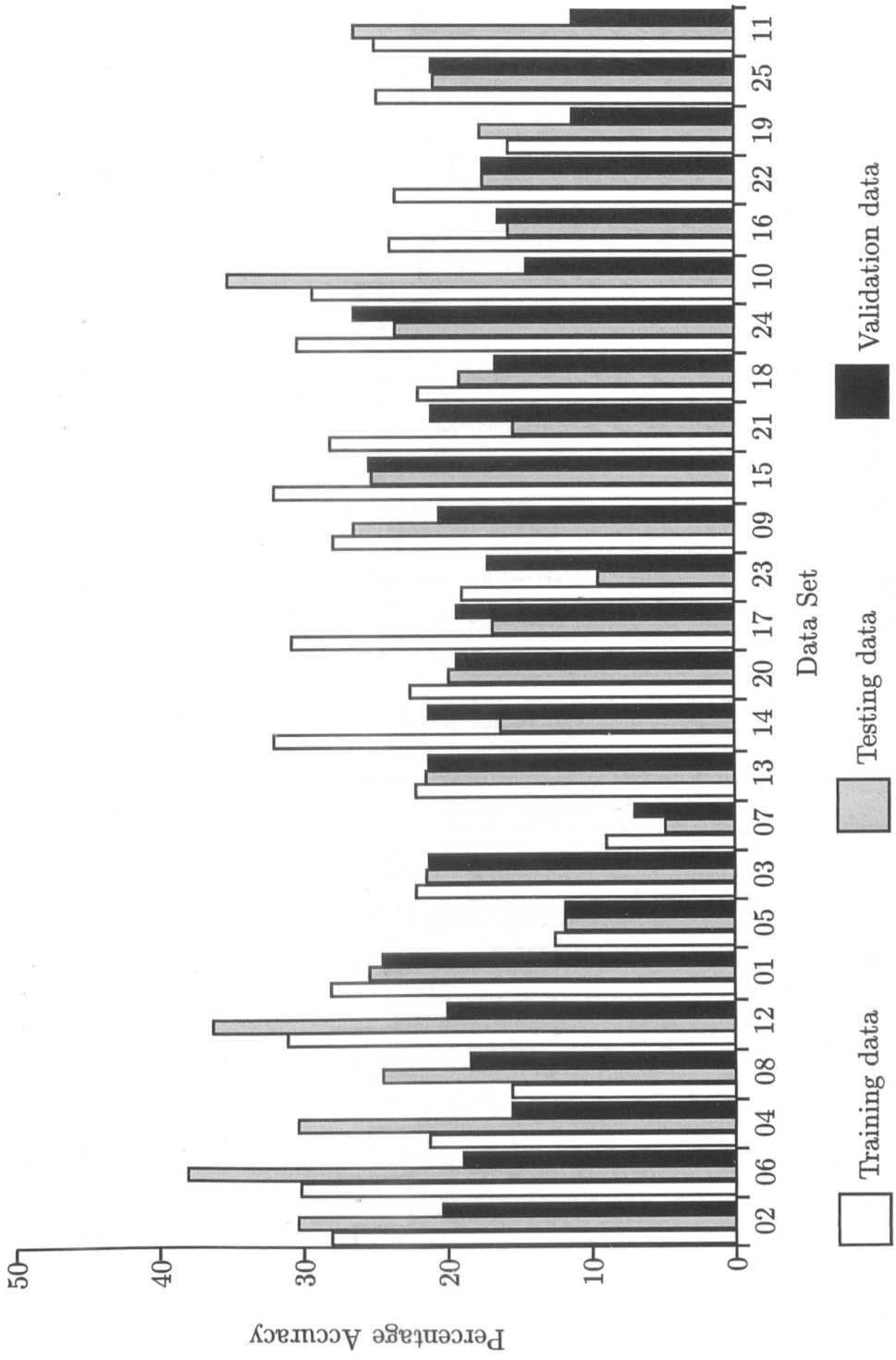


Figure 6.4 Eye colour data set comparison using an RBFN

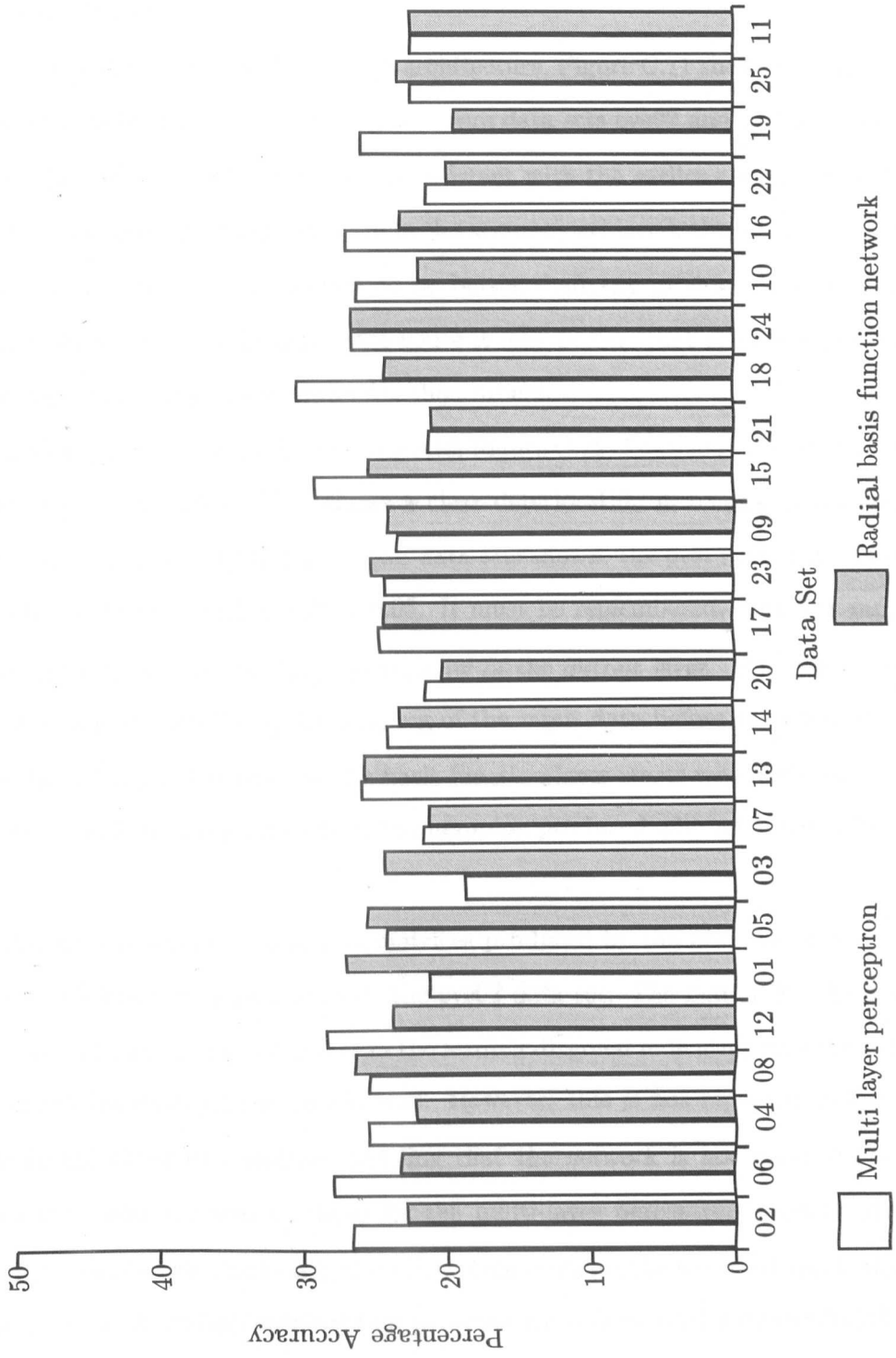


Figure 6.5 Eye colour data set comparison between network types

a marginal advantage over the radial basis function network with the problem of eye colour classification.

Now considering the RBFN network architecture, Figure C.11 shows the comparison between the different network sizes. Once more data sets eye02 and eye09 are used for making the comparison in order to be consistent with the earlier comparisons. While both data sets exhibit a large variation in the accuracy achieved, there are no particular network sizes that show themselves to be better than any others. Each line on the graphs represents a given learning rate hence it may be said that certain learning rates are giving significantly lower accuracies than others.

Turning our attention to the learning rate, Figure C.12 shows the accuracies achieved as learning rate is varied. This shows a clear deterioration in results as the learning rate is increased above 0.05. For the two data sets shown, the best results are obtained with learning rates of either 0.02 or 0.05. It must be remembered, that this variation in learning rate is only affecting the training of the output layer. The basis function layer will have already learnt its mapping of the input data before the training of the output layer begins. However, as the basis function layer starts with different random weights for each training simulation, the mapping produced will not be identical each time.

Table 6.5 presents the confusion matrices produced by the network that returned the best validation data results using the eye02 data set. The matrix for the training data shows a large number of entries in the leading diagonal indicating that the network has learned the data set reasonably well. However, this is not repeated to the same extent in the other two matrices showing that the network is not generalising well. Comparing these matrices to those for the multi-layer perceptron network given in Table 6.3 a similar distribution is observed. This confirms the expected result that the performance of the radial basis function networks are also controlled by the distribution of data within the input space.

As with the multilayer perceptron, comparison may be made between the colour representations used for the input data. Table 6.6 shows the confusion matrices generated when validation data is applied to the networks which return the greatest validation

Table 6.5 Confusion matrices for RBF network classifier using the eye02 data set

Training Data

undetermined	51	55	70	16	38
brown	49	22	0	16	**342**
hazel	84	22	28	**360**	84
green	30	11	**378**	8	26
grey	44	**308**	14	32	18
blue	**207**	22	0	16	18
	blue	grey	green	hazel	brown

Testing Data

undetermined	12	18	10	0	6
brown	7	0	0	0	**45**
hazel	3	9	0	**45**	15
green	5	0	**0**	0	0
grey	2	**18**	10	0	0
blue	**28**	18	40	15	0
	blue	grey	green	hazel	brown

Validation Data

undetermined	98	121	169	108	64
brown	157	99	169	162	**392**
hazel	21	22	26	**81**	14
green	11	0	**13**	0	4
grey	102	**187**	78	144	56
blue	**133**	88	78	45	40
	blue	grey	green	hazel	brown

Table 6.6 Validation data confusion matrices for eye colour classification using an RBF network and different colour representation data sets

Hue Data (27.1%)

undetermined	51	55	65	36	76
brown	108	99	52	72	**232**
hazel	348	341	390	**423**	254
green	0	0	**0**	0	0
grey	15	**22**	26	9	8
blue	**0**	0	0	0	0
	blue	grey	green	hazel	brown

Grey Data (26.6%)

undetermined	126	132	78	18	56
brown	191	176	195	207	**394**
hazel	203	209	260	**315**	120
green	2	0	**0**	0	0
grey	0	**0**	0	0	0
blue	**0**	0	0	0	0
	blue	grey	green	hazel	brown

Red Data (25.3%)

undetermined	85	99	91	54	46
brown	299	308	286	234	**438**
hazel	45	55	39	**180**	34
green	26	11	**26**	27	24
grey	41	**33**	39	27	16
blue	**26**	11	52	18	12
	blue	grey	green	hazel	brown

Green Data (26.7%)

undetermined	77	132	39	45	46
brown	175	165	221	162	**368**
hazel	150	121	156	**252**	108
green	0	0	**0**	0	0
grey	0	**0**	0	0	0
blue	**120**	99	117	81	48
	blue	grey	green	hazel	brown

Blue Data (23.5%)

undetermined	67	66	52	54	44
brown	191	231	325	261	**406**
hazel	46	55	130	**153**	90
green	2	0	**0**	0	4
grey	0	**0**	0	0	0
blue	**216**	165	26	72	26
	blue	grey	green	hazel	brown

accuracy for each of the different colour data representations. Once more the best results for each eye colour class have been highlighted by surrounding the figure with a box. These matrices show a tendency for the networks to be biased towards one or two of the classes and it is noticeable that the green and grey classes have particularly poor identification rates. This problem was also apparent with the multilayer perceptron networks, indicating a general problem in using these data sets for identification of these eye colours.

Of the "best" results identified in these matrices, three of them are produced with the same data sets as in the multilayer perceptron case. That is, the hue data gave the best hazel eye classification, the red data gave the best green eye classification and the blue data gave the best blue eye classification. This may indicate that there is information in these particular colour representations that is particularly suited to identifying these given eye colour classes.

6.3 Further Neural Network Methods

None of the neural networks examined so far has achieved over 30% accuracy in determining eye colour when using a threshold of 0.2 on the output value. Since the "random number classifier" would give an accuracy of 20% this does not represent a huge improvement and is not a results that would allow such systems to be used practically. Other methods need to be considered here in order to establish the usefulness of neural networks in solving this problem.

6.3.1 Experimental Methods

Thus far, the neural networks used for eye colour classification have employed five output neurons, one per colour, giving a single network for the whole classification problem. An alternative approach could make use of a single network for each eye colour and then combine the outputs of these networks to establish the classification.

There are a number of benefits to this type of system. Each network is only trying to map a single output function rather than using a set of weights to produce a number of different, albeit related, outputs. Therefore the task to be performed by each network is

Table 6.7 File sizes for single eye colour data files

Data set group	Training Patterns	Testing Patterns
Blue eyes	859	135
Grey eyes	1619	180
Green eyes	1664	186
Hazel eyes	1643	188
Brown eyes	1122	140

simpler and each network has fewer free parameters to evaluate and should therefore do better. In addition, since different input data sets are available, this method presents the opportunity of using a combination of different input data sets. That is, one particular data set may be better matched to identifying a given eye colour than the others.

The data sets chosen for this work employed the 8×8 sample area that was used for data sets eye09 - eye13. These data sets performed well in the previous experiments, and being the smallest, give the fastest training times. However, for these simulations, the data sets were regenerated with single colour targets. That is, a set of data files was created whose target value indicated whether the eye in question was brown or not and another set for the blue eyes etc. Since the numbers of examples of each of these classes varies in the database, when the balancing operation is performed on these data sets, the resultant data sets vary in size. These sizes are shown in Table 6.7. All five groups of data sets are using the same number of unique patterns; it is simply the process of repeated inclusion of patterns in order to equalise the number of data examples that is causing the sizes to vary.

A series of nine multi layer perceptron networks were used in this problem. Each of the five data sets was used in classifying each of the five eye colours and once more each network training run was simulated five times for each of six different learning rates giving a total of 6750 network simulations. These networks effectively return to the two class case used for moustache and beard identification so the same thresholding methods were applied, i.e. with a threshold of 0.2, outputs greater than 0.2 representing class $\mathcal{A}$, those less than -0.2 representing class $\mathcal{B}$ and the rest undetermined. As these networks are distinguishing between two classes, the expected result from a classifier

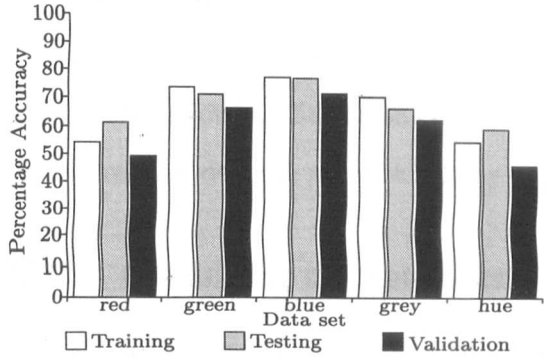
based on a random number generator is now 50% if there is no attempt to assign a degree of confidence to the result.

6.3.2 Results

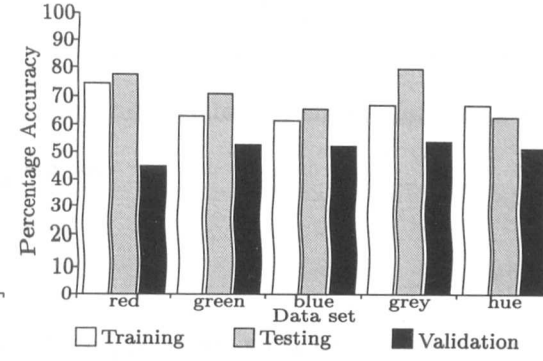
Since there are five different eye colours to classify, results are presented for each of these five colours. Figure 6.6 shows a comparison of the use of each data set for each eye colour. From all the simulations performed, a representative network was chosen for comparison of the data sets. The network used in this case is a multi layer perceptron with 20 hidden layer neurons and a learning rate of 0.02. Rather than use the number to identify a given data set, since there is only a single sample size being used in these simulations, the colour representation will be used to identify data sets. This figure shows that using this network architecture, certain eye colours are more easily identifiable than others. For example, the blue and brown eyes are most easily identifiable with the hazel being the most difficult. In addition it is clear which data sets most easily identify any given eye colour, for example, the blue data set gives the best identification rate with blue eyes and the red data set gives the best results for green eyes.

Considering network architecture, Figure C.13 shows how the results vary as the size of the hidden layer is altered. In this case, the grey data set was used and traces are plotted for each of the different learning rates used. The blue and brown eyes clearly stand out as those most easily identifiable and with the least variation in accuracy as the size of the network is varied. The other three cases show somewhat unpredictable behaviour. The accuracy figures appear to be dependent on the precise combination of network size and learning rate with no clear trends visible to indicate which network size will be optimal.

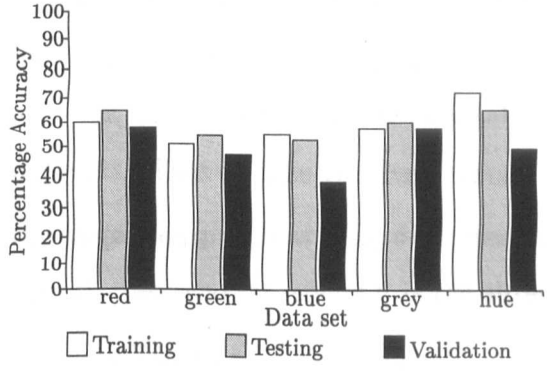
The plot of accuracy against learning rate shown in Figure C.14 shows that most of the variation in accuracy is caused by changing the learning rate used to train the network. In all cases, the lowest accuracy was achieved with the learning rate of 0.01 rising to the best results with one of 0.05, 0.1 or 0.2. Some networks showed a reduction in accuracy with the highest learning rate of 0.5, probably due to the large changes



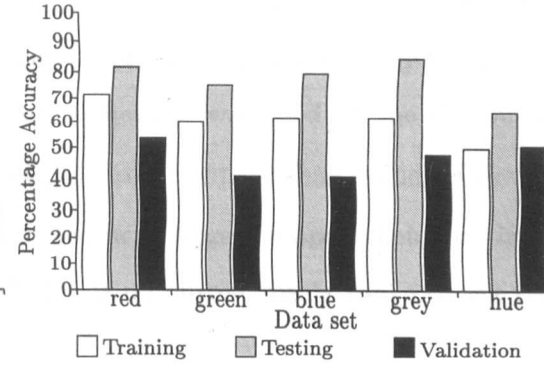
(a) Blue Eyes



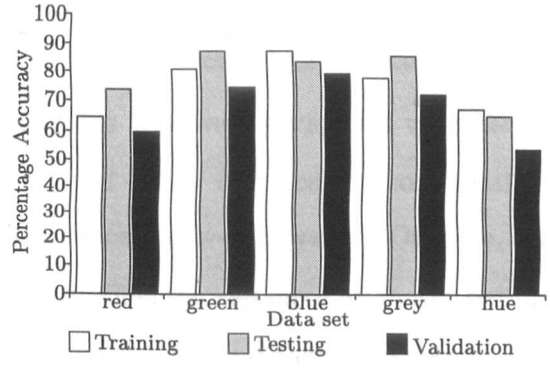
(b) Grey Eyes



(c) Green Eyes



(d) Hazel Eyes



(e) Brown Eyes

Figure 6.6 Comparison of data sets used in single eye colour classification with fixed MLP architecture

that are associated with a large learning rate making it difficult for the network to find the bottom of the global minimum on the error surface. As was seen in the plot of accuracy against hidden layer size, (Figure C.13), the accuracy of the results when classifying blue and brown eyes are both high and vary little as network parameters change thus showing that these classifications are represented by simple functions that do not require delicate network tuning to achieve.

6.3.3 Combining the networks

In each of the preceding cases, results have been presented with just the accuracy of identifying a single eye colour out of the five possible colours. In order to provide a complete solution, these networks need to be combined to form a single system which identifies the colour of a given eye. The simplest method by which this may be achieved is to use the same kind of "winner takes all" system as was used on the five output networks. That is, take the five network outputs and compare their values. Provided the largest output is at least θ greater than the next largest output then that is the correct classification, where θ is the threshold. If no output meets that criteria then the classification is 'undetermined'.

In combining the outputs of five independent networks, decisions have to be made as to which five networks are to be combined. This is where advantage may be taken of the use of different input data representations to classify different eye colours. Looking at the number of simulations that were performed with the single colour recognition networks, there are a total of 4.48×10^{15} possible combinations network outputs that could be performed to produce the overall system. Of these, six different combinations have been chosen based on different measures of "best" performance. The data combinations used are listed in Table 6.8 and they were created according to the following procedure:

All the networks used in any given eye colour classification were considered and the networks that gave the highest accuracy when presented with each of the training, testing and validation data were identified. A group of data sets, balanced according to the number of examples in each of the five classes, was applied to these networks and the

Table 6.8 Data sets produced by combining single eye colour network responses

Data set	Single eye colour networks used
eyec01	Best Testing
eyec02	Best Validation
eyec03	Best Training
eyec04	2nd Best Testing
eyec05	2nd Best Validation
eyec06	2nd Best Training

results combined to produce three data sets. That is, one from combining the outputs of the networks that had the greatest training accuracy, one from the networks with the greatest testing accuracy and one from the networks with the greatest validation accuracy. These three data sets are referred to as "Best Training", "Best Testing" and "Best Validation" in Table 6.8.

This process was then repeated with the networks that gave the second highest accuracy when presented with each of the three input data sets to produce three further combined data sets. These are referred to as "2nd Best Training", "2nd Best Testing" and "2nd Best Validation" in Table 6.8.

In order to obtain the results for a "winner takes all" system, these data sets were passed through the same routine used to analyse results from the five output networks (see Algorithm 6.1 in 6.1.2). The results from this process are presented in Figure 6.7. The "best" results in this set are obtained using the data set produced from combining the "Best Validation" data sets, i.e. data sets eyec02 and eyec05. However doing so, alters the meaning of the validation data sets. It has now become associated with the training process as measures taken from use of that data set have been used to calculate the best combination of data sets. If the validation data is to remain strictly for validation use only then this data set must be discounted in which case the performance of the overall system is much lower and best results are achieved using the data set constructed from the "Best Training" data sets.

It is noticeable in this graph that the rule used to combine single eye colour network results to produce the data sets for these tests is reflected in the results obtained. That is, the two data sets constructed using the "Best Testing" rule (eyec01 and eyec04) both give much higher results with testing data than the others. Similarly the two data sets

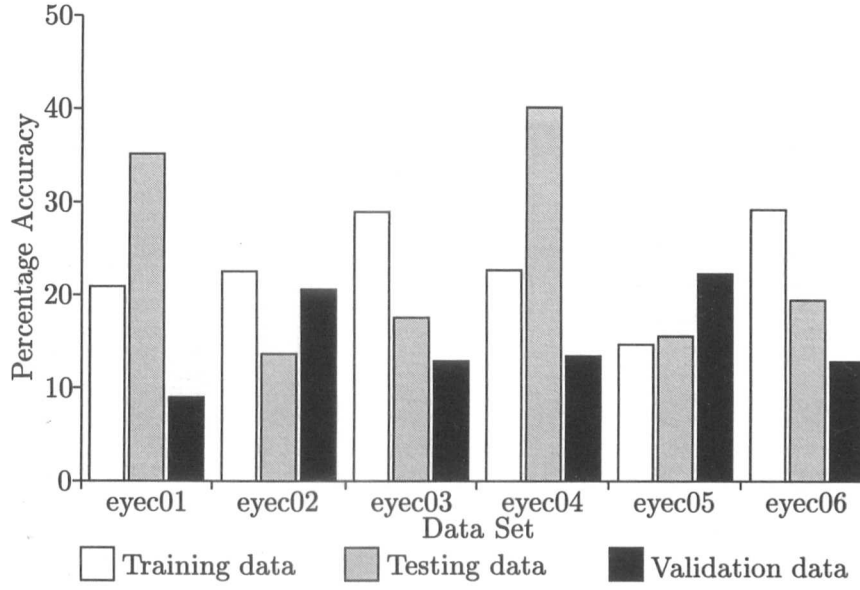


Figure 6.7 Results from combining the outputs of five single eye colour networks using a "Winner Takes All" algorithm.

constructed with the "Best Training" rule give greater accuracy with the training data than the other sets.

Instead of just using a simple "winner takes all" approach to combining the results from the five independent networks, a further neural network can be used. In this case, the network requires five inputs and five outputs since it is taking output from five networks and producing an output indicating which of five classes the eye belongs to. The number of hidden layer neurons is varied as usual. The same nine hidden layer sizes were used as were applied to identifying the individual colour classes. Given the six data sets, ten hidden layer sizes, six learning rates and five simulations of each network, there were a total of 1800 network simulations. Taking a network with 20 hidden layer neurons and a learning rate of 0.05, the six data sets may be compared as shown in Figure 6.8. This system produced the best results using the eyec01 data set, this result showing an improvement on the comparable result from the "winner takes all" results. However the other five data sets all performed worse with this neural network classifier than with the "winner takes all" analysis with a drop of between 5% and 15% in their validation accuracy.

It can be seen that, with the exception of eyec01, all the other data sets returned

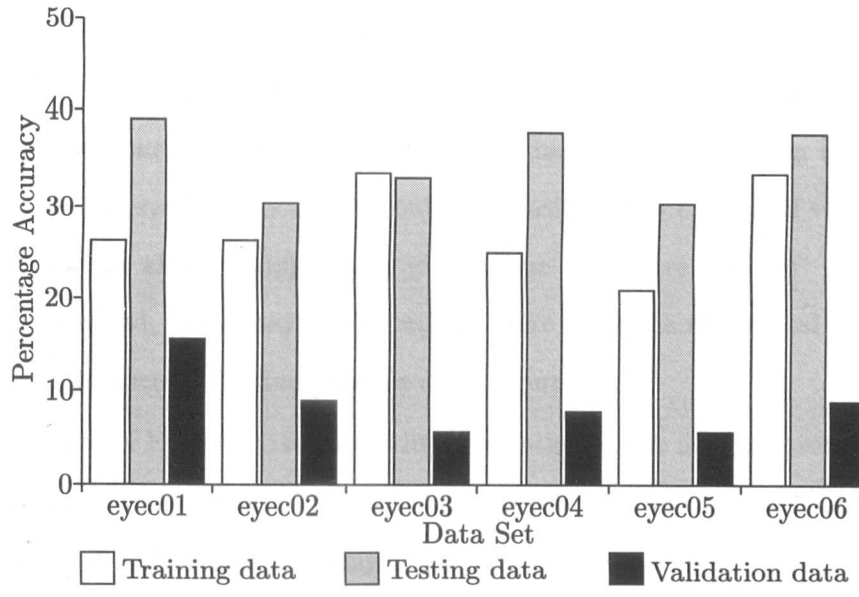


Figure 6.8 Eye colour data set comparison - combining single colour networks

significantly poorer results from testing with the validation data than with the training and testing data. This would indicate that the networks were learning the mapping needed for the training data but were failing to generalise beyond this. These results are however taken from just a single network configuration out of the many that were simulated; others may have given better results and will be considered later when all the classification methods are compared.

Figure C.15 which compares the different sizes of hidden layer and Figure C.16 which compares the different learning rates both have the results from training, testing and validation data split into three clear "bands". The validation data always returns the lowest of these results as would be expected. The order of the other two is dependent on the data set being used; with the two data sets based on the "Best Training" measure (eyec03 and eyec06), the training data gives the better results whereas with the other two data sets, testing data yields higher accuracies.

The network systems being simulated here may be compared to encoder or compression networks. These systems consist of networks with identical numbers of input and output neurons and usually a smaller number of hidden layer neurons. The network is taught in such a manner that the outputs should be identical to the inputs,

the purpose being to compress the data through the smaller number of hidden layer neurons. Theoretically, an encoder network with 2^n inputs and outputs needs only n hidden layer neurons, if neurons can only have values of zero or one on their outputs. In the case of this system, the inputs could be considered as corrupted versions of the output as they are already supposed to give a "one out of five positive" value. As the input are corrupted, there may be a need for more than the theoretical three hidden layer neurons to encode the five possible desired outputs.

Examination of Figure C.15 reveals little variation in the accuracy returned as the size of hidden layer is varied. Many of the data sets do show a significant reduction in accuracy with only two hidden layer neurons, but with three or more, the results are fairly consistent. This increase in accuracy with three hidden neurons is consistent with the above theory that the system is effectively an encoder system. The absence of any further significant increase in accuracy would suggest that the 'corruption' in the inputs that can be redressed is already being realised by these three hidden neurons and no further improvement in response can be made.

Figure C.16 shows a small reduction in accuracy when the networks are trained with the largest learning rate indicating that the larger changes in the values of network weights, caused by the higher learning rate, are too large for the networks to find the global minimum in the error surface.

Figure 6.9 presents the results for the best network for each of the six data sets. It is still the data sets constructed using the "Best Validation" test (eyec02 and eyec05) that return the best results. These are followed by the "Best Testing" sets (eyec01 and eyec04) with the "Best Training" (eyec03 and eyec06) giving the poorest results.

6.3.4 Comparing the classification methods

In order to evaluate whether this alternative method of using a single network for each eye colour and then combining the outputs has performed any better than the previously examined methods, comparison must be made between Figure 6.9 and Figure 6.5. The latter uses many more data sets than have been applied in the alternative network techniques so comparison of all entries in Figure 6.5 is not appropriate. The most

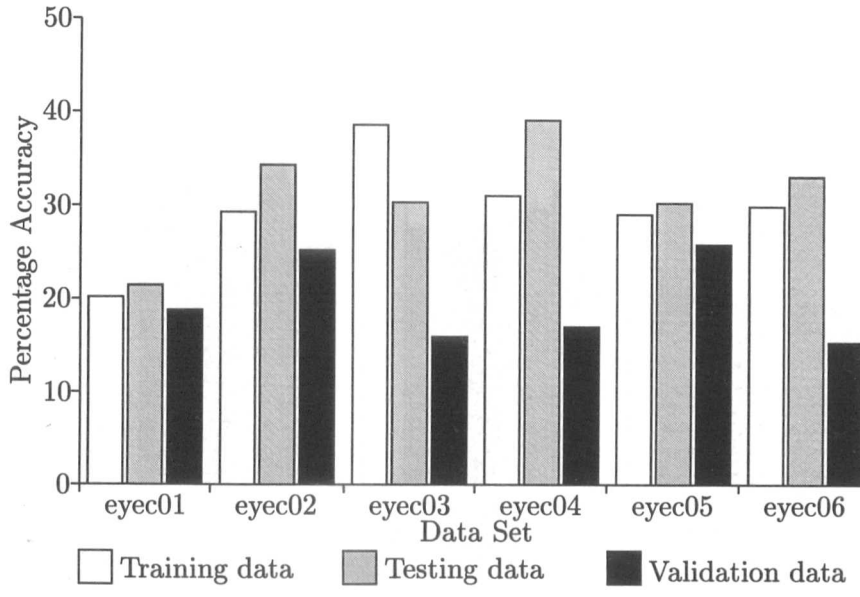


Figure 6.9 Eye colour - combining single colour networks - best results

similar data sets are eye09, eye10, eye11, eye12, and eye13 as these have 16 elements per input vector in the same manner as the data sets used to train the single eye colour networks.

The overall comparison of the neural network methods is presented in Figure 6.10. This plot presents the best validation result acheived with each of the data 'types' used for each network type. I.e. each of the five colour representations are presented for the MLP and RBF networks and each of the six data sets are presented for the "Multinet" system. For this reason, the shading of the bars does not have specific meaning, it is simply to distinguish the different bars. It can be seen that the best results are achieved with the original MLP design and similar performance resulting from the RBF networks. The "Multinet" approach, that is one network per eye colour, performs noticeably worse than the other two methods with the exception of the results from eyec02 and eyec05. However as has already been discussed, the use of these particular results invalidates the meaning of the validation data as it was used in the selection process in combining the outputs of the individual eye colour networks.

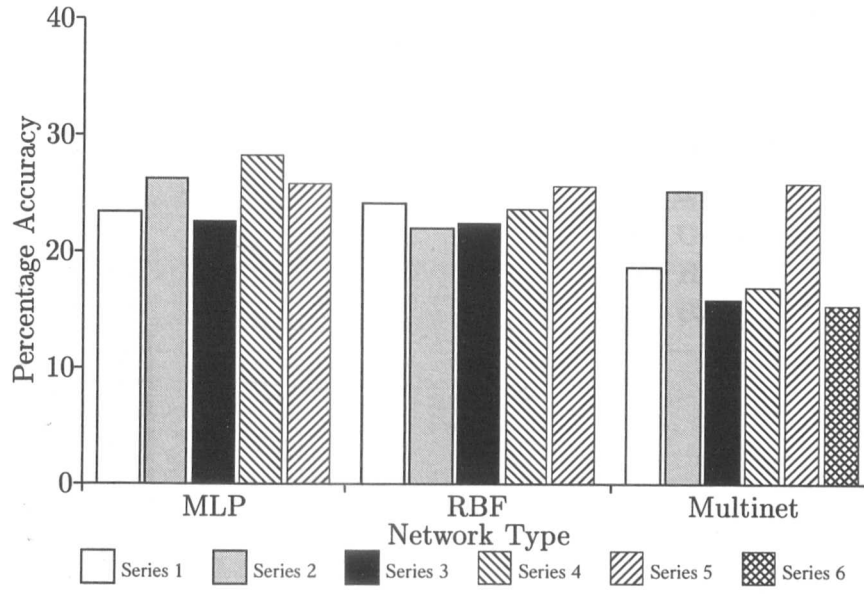


Figure 6.10 Eye colour - comparing the three network approaches

6.4 Further Data Analysis

Given the disappointing results acheived in attempting to classify eye colour, further analysis of the data is necessary to establish reasons for these results. Using Matlab, a form of probability density function (PDF) analysis was perfomed on the eye data sets. As mentioned in 4.2.1, the hue colour representation is the one which represents the underlying colour of any given pixel and therefore should be the one which most clearly distinguishes between the different eye colours.

The hue data sets were therefore presented to a Matlab program which stored against each eye colour the number of occurances of each hue in the images presented. For plotting, the hue data has been scaled from the 0° to 360° range to a range of 0 to 1. These have then been plotted and two such plots are given here.

Figure 6.11 shows the results when the PDF analysis is performed on data set eye01. This is the hue data set that presents the most accurate information about the eye colour as it uses 1×1 sub-sampling to produce the final data. This plot clearly shows practically no separability between the classes with respect to the hue data.

Figure 6.12 presents the results of PDF analysis on data set eye13. This data

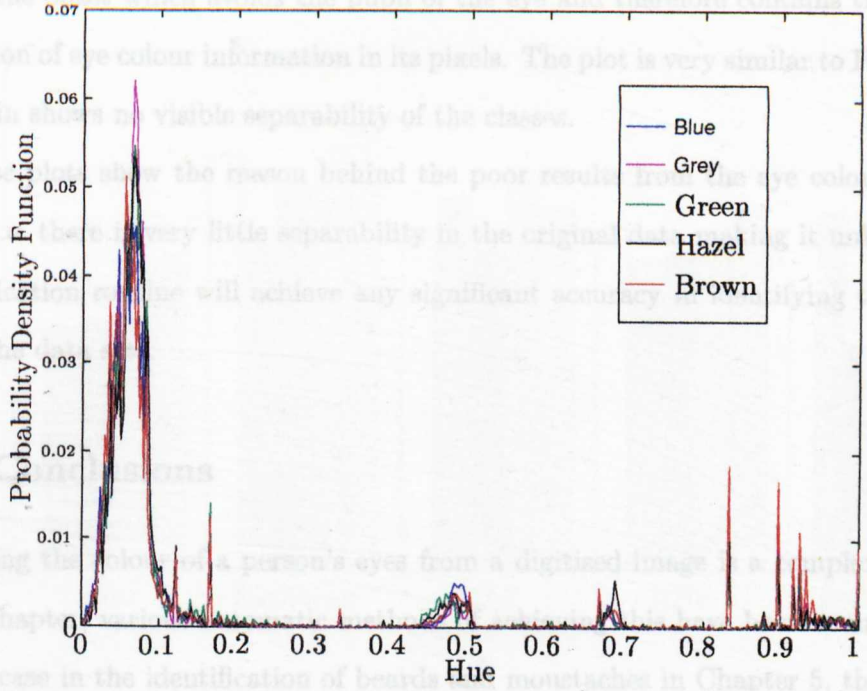


Figure 6.11 Eye colour - PDF on eye01 data

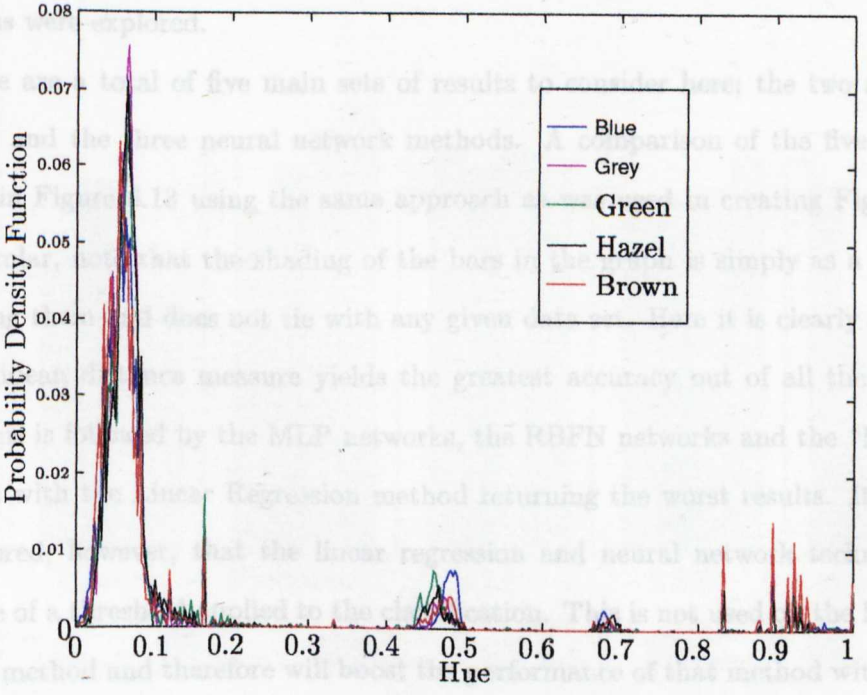


Figure 6.12 Eye colour - PDF on eye13 data

set has the offset which avoids the pupil of the eye and therefore contains the highest proportion of eye colour information in its pixels. The plot is very similar to Figure 6.11 and again shows no visible separability of the classes.

These plots show the reason behind the poor results from the eye colour classification, i.e. there is very little separability in the original data making it unlikely that a classification routine will achieve any significant accuracy in identifying the classes within the data sets.

6.5 Conclusions

Classifying the colour of a person's eyes from a digitised image is a complex process. In this chapter, various automatic methods of achieving this have been examined. As was the case in the identification of beards and moustaches in Chapter 5, the number of parameter combinations that could be used in the neural network simulations are huge. However the combinations used here covered a broad range of possible values to show the potential that is possible in using neural networks. Again the most popular variations were explored.

There are a total of five main sets of results to consider here; the two statistical methods and the three neural network methods. A comparison of the five methods is given in Figure 6.13 using the same approach as was used in creating Figure 6.10. In particular, note that the shading of the bars in the graph is simply as a means of separating them and does not tie with any given data set. Here it is clearly seen that the Euclidean distance measure yields the greatest accuracy out of all the methods used. This is followed by the MLP networks, the RBFN networks and the "Multinet" networks with the Linear Regression method returning the worst results. It must be remembered, however, that the linear regression and neural network techniques all make use of a threshold applied to the classification. This is not used on the Euclidean distance method and therefore will boost the performance of that method with respect to the others.

Figure 6.14 is a repeat of Figure 6.13 but with the threshold functions removed from all of the classification methods. This gives the most accurate comparison between

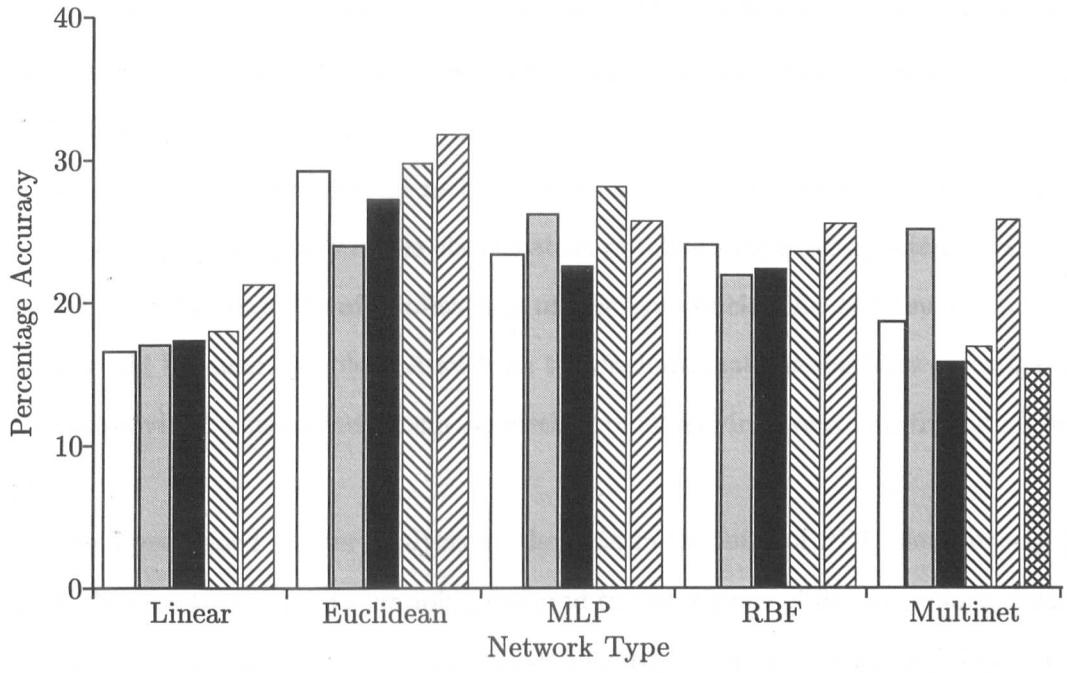


Figure 6.13 Eye colour - comparing all classification approaches

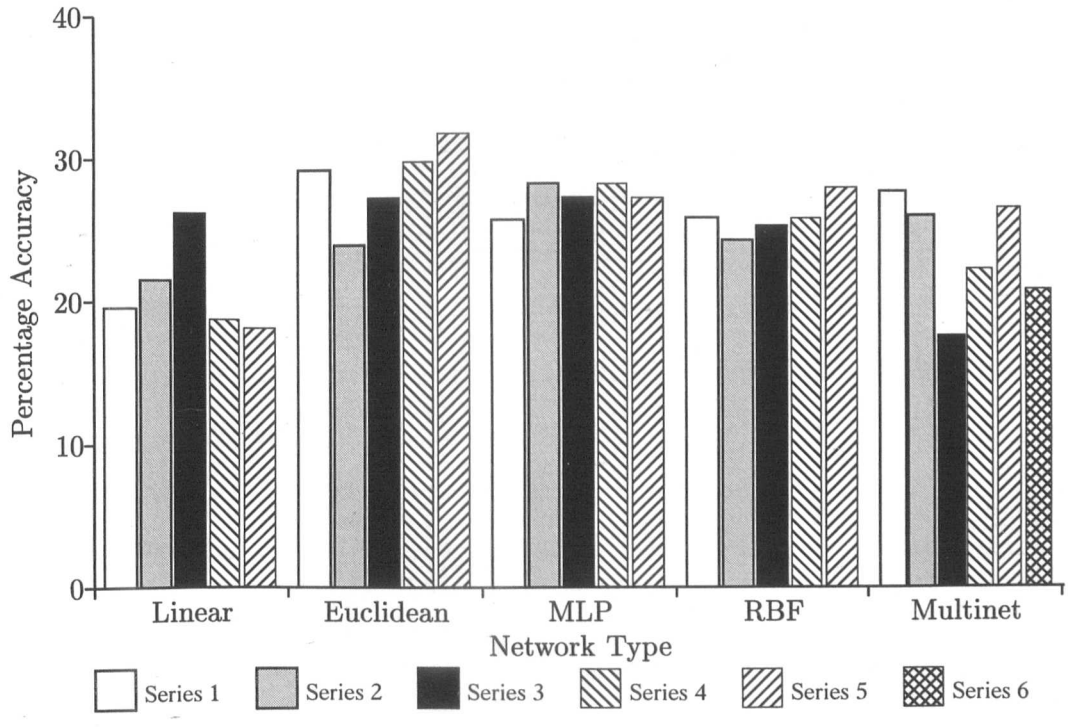


Figure 6.14 Eye colour - comparing all classification approaches (no thresholds)

the different techniques. Removal of the thresholds has increased the classification accuracy of the relevant techniques but the overall order has remained the same, with the Euclidean distance method performing best.

The fact that the best of these methods achieves only just over 30% accuracy shows something of the complexity of the classification problem that is being attempted here. Obviously for the majority of applications of this type of classification, such a performance would be unacceptable and we have to conclude that the methods investigated have not provided a sufficiently accurate method of classifying eye colour from digitised images.

Having performed further analysis on the data in the form of PDF plots, it may in fact be the case that the data mapping required to classify one of five eye colour classes from the supplied digitised images is not possible due to the lack of separability in the data.

Chapter 7

Conclusions

If artificial neural networks, as described in Chapter 2, are considered to be based on the workings of the human brain, one might expect that an artificial neural network which is designed for examining an image of a face would be able to give a similar description to that produced by a human observer. However, the artificial neural networks that are currently in use are only a very simple representation of the biological system and in reality, our understanding of the biological neural system is not sufficiently complete for such systems to be developed. Rather ANNs should be seen as an alternative tool for data processing work with particular properties as discussed in Chapter 2. It is also the case that the biologial neural systems inherently used by humans for the purpose of describing facial features are many orders of magnitude larger than the computer simulations that are typically used in artificial neural systems.

Therefore in this work, the use of neural methods to extract psychological descriptions is not significanct in terms of relative scale or complexity. We have typically simulated a few hundred neurons here whereas the human brain contains 10^{11} neurons. Nevertheless, it is the particular powers of the artificial neural network acting as a classifier that are being used rather than any link with biological systems.

The two sets of facial features chosen for classification in this thesis can be said to represent the two extremes of complexity within the problem of facial feature classification and this is reflected in the relative accuracy of the results achieved for each feature.

Beard and moustache identification is a simple process for a human observer. The features are usually obvious from a visual inspection of the picture and therefore one would expect that an image analysis system designed to analyse the appropriate areas of the image would be easily able to identify the presence or absence of the features in question. This is indeed what has been found as a result of this work. Results presented in Chapter 5 show that simple linear statistical classifiers are able to achieve results as high as 80% when examining results using testing data. These were then surpassed by the neural networks which produced results of nearly 100% accuracy. The improved performance of the neural networks indicates that the subdivision between the two classes of images is more complex than can be determined by the simple statistical methods and shows the benefits that may be gained by using neural networks for classification. The non-linear capabilities of the neural system were better suited to describing the mapping between image and classification than a simple distance classifier.

Visual inspection of any group of eyes will reveal the complex patterns of colours that may be observed. The FRAME encoding reduces this to just five colour categories, therefore a classification process is required that can establish which of the five categories any given eye belongs to. Given the wide range of possible eye colour patterns, this classification is never going to be precise, that is, a 100% accuracy should not be expected, or in fact, desired. This is further placed in context when we recall that the original "correct" classifications were allocated by a group of human jurors and therefore is subject to human error. The level of difficulty of the classification problem was shown by the inability of PCA and Kohonen techniques to show any separation between the five classes of colour. Some grouping of the data into classes must be present as the Euclidean distance classifier was able to return accuracies above 20%, even when examining validation data, which is the accuracy that should be returned by a random classification of one out of five classes.

Our analysis shows that in the eye colour classification problem, it is the Euclidean distance method that achieves the best results. There is one caveat in this statement which is that the Euclidean distance method involves no thresholding mechanism

whereas the other methods all have a threshold set to reduce the effect of noise. Therefore for the most accurate comparison, the threshold functions need to be removed. This would not be the case in a real application as the noise reduction properties would be required; the presence of an "unclassified" class is helpful in eliminating classifications where the probability of the result being correct is low. Having noted this, Figure 6.14 shows that the Euclidean distance measure still slightly outperforms the other methods in classifying eye colour.

These results and the PDF analysis subsequently performed on the eye colour data leads to the conclusion that the colour of an eye can not easily be automatically determined from a colour image of the eye. Alternative classification methods will need to be examined in order to find a solution to this problem almost certainly involving some different data representation from the colour models that have been used in this work.

One of the difficulties in determining the success or otherwise of this work is that there are no real benchmarks which can be used for comparison. In addition, there are no standard libraries of facial images available with the "FRAME" descriptive measures so the work is limited to the use of the 1000 faces supplied by the home office. Consequently the results of this work have to be considered in the light of what would be required were the system to be implemented in a "real world" situation. It is these considerations that result in the conclusion that the work has been successful in detecting the presence or absence of beards and moustaches but not in the case of classifying eye colour.

7.1 Future Work

In examining the future possible direction of this work it is helpful to look again at the objective of the overall research which this current work fits into. The aim is to develop a system that is capable of automatically extracting a set of descriptive features from a digitised photograph of a face. On a larger scale, this work is to fit into a system where these descriptions are used as an index into a set of photographs stored in a 'database'.

As it stands, the work presented in this thesis has only examined three of the

fifty descriptive measures present in the original FRAME database. Therefore there is plenty of scope for additional work in classifying other descriptive features. It is also worth noting that most of the remaining features are measured on a scale from 1 to 5 rather than having discrete values. This brings up the question of data representation for real valued numbers within a neural network. [28] explores various different data representations that may be used.

When considering these other features, the distinction made in Table 4.1 and Table 4.2 must be taken into account. That is, some of the features may be measured as physical distances or areas from the image and therefore will not need any special classification methods such as ANNs.

Along with the classification of eye colour that has been examined in this thesis, there are other colour based features to consider such as "hair colour" and two of the features relating to complexion. Other "groups" of features may be defined as those that are related to *shape* such as "Shape of face", "Eyebrows" and "Forehead", and those related to *texture* such as the "unlined-lined" and "clear-blemished" complexion measures. Within each of these groups, the features in question are likely to require a similar solution to the classification problem, therefore the techniques used in this work in classifying eye colour could also be applied to the classification of hair colour. Specific techniques may be used that relate to the nature of colour vision, for example, Usui *et al.*[83] have proposed a neural network model of colour vision. Such methods could be used to tailor the neural system more accurately to the desired classification.

However, some of these other features provide a much more complex problem in terms of defining the area of image to consider. For example, locating a consistent area of image that contains hair is probably not possible due to the wide range of different hair styles that may be observed. Therefore an alternative method will need to be devised in order to make it possible to locate the hair in the image consistently before attempting to classify the colour in question.

Related to the work in this thesis is that of automatically identifying the location of the feature points on the facial image. The set of points used for the work in this thesis were manually entered therefore since one of the goals of the research is the production

of a completely automated system, this area needs to be addressed in addition to the feature classification. In addition, a method of locating these feature points is required for calculating the descriptive measures listed in Table 4.1. Chapter 3 has described a number of facial recognition techniques, some of which make use of physical positions on the face[7][36][90]. The methods used in these recognition techniques may be adapted for the purpose of locating some of the feature points that are needed in the descriptive measure work performed here.

Various methods of locating feature points have been discussed in Chapter 3 and one of these could be suitably modified for use in this instance. It is worth remembering that while there are some 37 feature points shown in Figure 4.1, not all of these may be required in the final solution. A point will only be required if it is needed for identifying a given area of the image for examination or if it is used in calculating one of the "physically derived" measures.

Both the feature point location methods and those used for feature classification will need to look further at the effects of lighting on the image quality. In this thesis, certain images were rejected as the reflections from glasses were too much to allow proper examination of the eye, or the subject was wearing glasses with coloured lenses. For the former case, consideration will have to be given to the methods of controlled lighting that would be available in the 'live' environment. Consideration of the latter would have to be built into the procedure for taking photographs for analysis. It may have to become a requirement that any glasses being worn have clear lenses.

A further area of future work is the consideration of alternative specialised artificial neural network architectures. These could include networks that are specifically designed for image processing work such as the neocognitron[20] which has been successfully applied to handwriting recognition or modifications to the backpropogation algorithm such as those done by Anand *et al.*[2]or Pugmire *et al.*[63] which deal with imbalanced training data sets such as those that were experienced in this work. Simple logical extentions of the work performed here would include the use of a supervised network layer on the output of the SOM networks presented in Chapter 4 which it is estimated would produce very high results for the simple classifications. This type of

technique may prove useful in other feature classifications.

An altogether different area that is worth further consideration is the descriptive measures themselves. While research on such topics is outside the scope of this thesis, it is noticeable that none of the 50 measures makes any mention of the ears and this leads to the question as to whether the list of measures themselves needs re-assessing. In addition, some of the features may be easily altered, such as hair length and colour or the presence or absence of glasses. Therefore certain descriptive features may be seen as less reliable that others when using them as an index into a library of images.

Having stated, in Section 1.1, that all faces are made up from the same basic components, is has become apparent in the study of how to classify facial features that there is great variation in the makeup of the average face. In performing feature classification it is this variation that has to be both measured and compensated for. That is, when classifying a feature, one has to measure it in comparison to the scale of variation that is possible, but at the same time, locating the feature in question within the image will require some compensation for the movement and variation within other features in the face. The most obvious example of this is any classification associated with hair. The wide variety of hair styles that may be observed mean that the "rules" for determining the position of hair are very vague and on some full face images there may be difficulty in finding a sufficiently large area of hair to use to classify the colour. Certain features such as these may well require the use of side profile images in addition to the full face images used in this thesis, thus increasing the prerequisites for the classification process.

This thesis presents a first attempt in the design of a system to automatically classify facial features from digitised images. Certain areas of this work have proved possible within the limitations of the methods investigated, while others will require the consideration of different methods, possibly more tailored to the particular feature in question.

Appendix A

Derivation of f'

The logistic activation function $f(a)$ is defined as:

$$o = f(a) \triangleq \frac{1}{1+e^{-a}} \tag{A.1}$$

so

$$f'(a) = \frac{do}{da} = \frac{d}{da}\left(\frac{1}{1+e^{-a}}\right) \tag{A.2}$$

Let

$$u = 1 + e^{-a} \tag{A.3}$$

hence

$$\frac{du}{da} = -e^{-a} \tag{A.4}$$

Now by the chain rule

$$\frac{do}{da} = \frac{do}{du} \times \frac{du}{da} \tag{A.5}$$

(A.1) and (A.3) give

$$o = \frac{1}{u} \tag{A.6}$$

leading to

$$\frac{do}{du} = -\frac{1}{u^2} \tag{A.7}$$

So from (A.5)

$$\begin{aligned} f'(a) &= -\frac{1}{u^2} \times (-e^a) \\ &= -\frac{1}{(1+e^{-a})^2} \times -e^{-a} \\ &= \frac{e^{-a}}{(1+e^{-a})} \end{aligned} \tag{A.8}$$

which can be written

$$\begin{aligned} f'(a) &= \frac{1}{1+e^{-a}} \times \frac{1+e^{-a}-1}{1+e^{-a}} \\ &= \frac{1}{1+e^{-a}} \times \left(\frac{1+e^{-a}}{1+e^{-a}} - \frac{1}{1+e^{-a}} \right) \end{aligned} \tag{A.9}$$

So from (A.1)

$$f'(a) = f(a) \times (1 - f(a)) \tag{A.10}$$

A similar derivation can be used for the hyperbolic activation function:

$$o = f(a) \triangleq \frac{2}{1+e^{-a}} - 1 \tag{A.11}$$

so

$$f'(a) = \frac{do}{da} = \frac{d}{da} \left(\frac{2}{1+e^{-a}} - 1 \right) \tag{A.12}$$

Again using (A.3) we now get

$$o = \frac{2}{u} - 1 \tag{A.13}$$

giving

$$\frac{do}{du} = -\frac{2}{u^2} \tag{A.14}$$

Using (A.5) and (A.4) gives

$$f'(a) = -\frac{2}{u^2} \times (-e^{-a})$$

$$= -\frac{2}{(1+e^{-a})^2} \times (-e^{-a})$$
$$= \frac{2e^{-a}}{(1+e^{-a})^2} \tag{A.15}$$

which can be written as

$$\begin{aligned} f'(a) &= \frac{2+2e^{-a}-2}{(1+e^{-a})^2} \\ &= \frac{-2}{(1+e^{-a})^2} + \frac{2e^{-a}+2}{(1+e^{-a})^2} \\ &= \frac{-2}{(1+e^{-a})^2} + \frac{2}{1+e^{-a}} \\ &= \frac{1}{2} - \frac{2}{(1+e^{-a})^2} + \frac{2}{1+e^{-a}} - \frac{1}{2} \\ &= \frac{1}{2}\left(1 - \frac{4}{(1+e^{-a})^2} + \frac{4}{1+e^{-a}} - 1\right) \\ &= \frac{1}{2}\left(1 - \left(\frac{2}{1+e^{-a}} - 1\right)^2\right) \end{aligned} \tag{A.16}$$

So from (A.11)

$$f'(a) = \frac{1}{2}\left(1 - f(a)^2\right) \tag{A.17}$$

Appendix B

Conversion of RGB colour values to other representations

Algorithm B.1 taken from [19] may be used to convert pixel colours expressed in RGB notation to HSV notation. It assumes that the *red*, *green* and *blue* values are given in the range $[0, 1]$, *hue* is required in the range $[0, 360]$ and *saturation* and *value* are required in the range $[0, 1]$. Conversion to CMY is achieved by simple calculation using

$$cyan = 1 - red \tag{B.1}$$

$$magenta = 1 - green \tag{B.2}$$

$$yellow = 1 - blue \tag{B.3}$$

again assuming values for the colour components in the range $[0, 1]$.

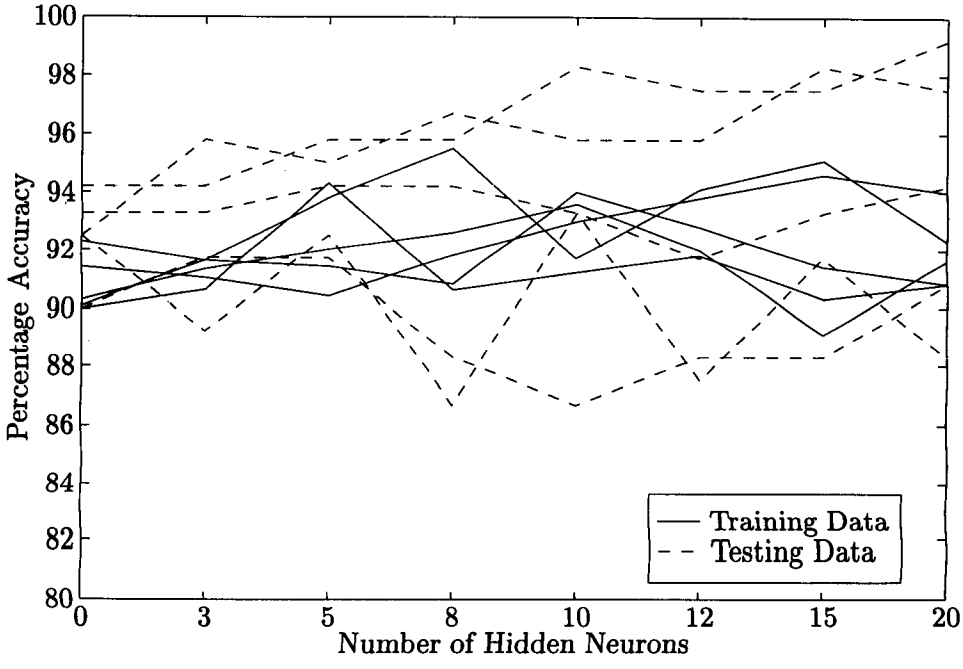
Algorithm B.1 RGB to HSV conversion

```
max ← MAX(red, green, blue)
min ← MIN(red, green, blue)

value ← max

if max ≠ 0 then
  saturation ← (max - min) / min
else
  saturation ← 0
end if

if saturation = 0 then
  hue ← UNDEFINED
else
  delta ← max - min
  if red = max then
    hue ← (green - blue) / delta
  else if green = max then
    hue ← 2 + (blue - red) / delta
  else
    hue ← 4 + (red - green) / delta
  end if

  hue ← hue × 60
  if hue < 0 then
    hue ← hue + 360
  end if
end if
```

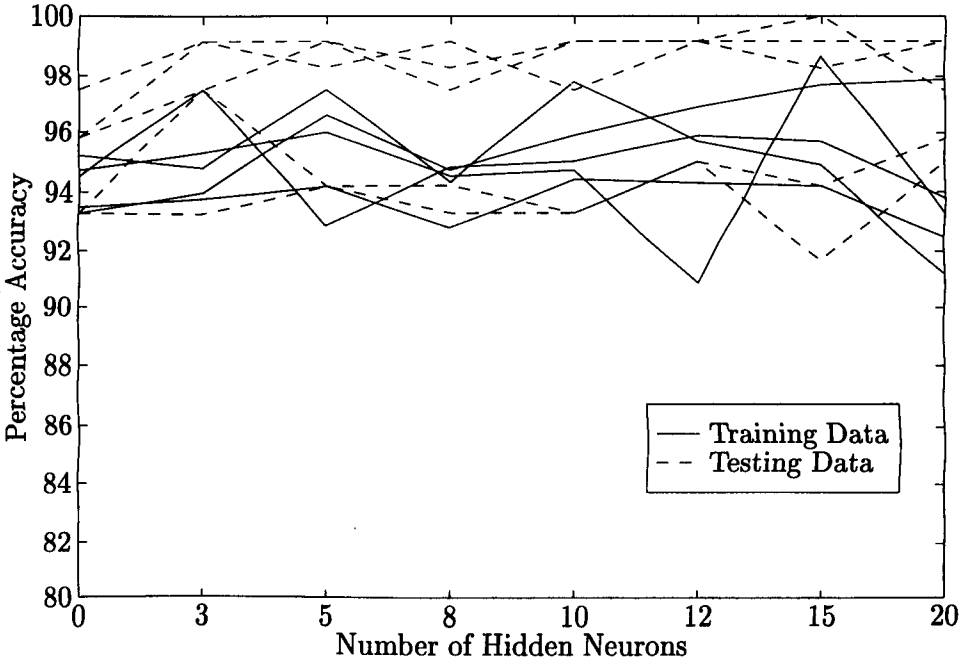
Appendix C

Graphs Used in Evaluating Training Parameters

The graphs in this appendix are those referred to in Chapters 5 and 6 when considering the number of hidden layer neurons and the learning rates used to produce optimal results. They are presented in the order in which they are referred to in the chapters.

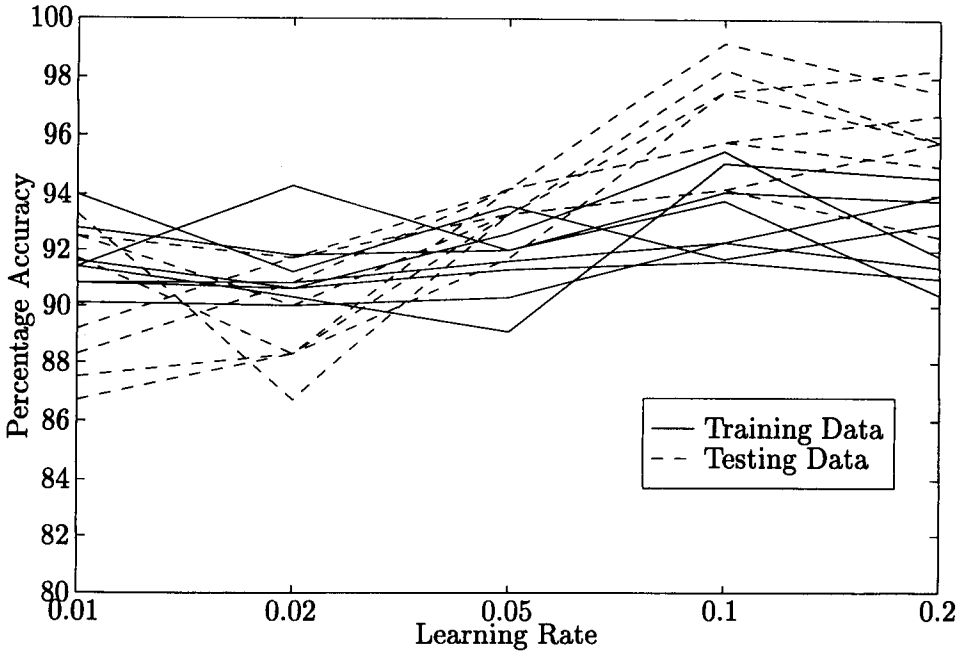


(a) Data set m01

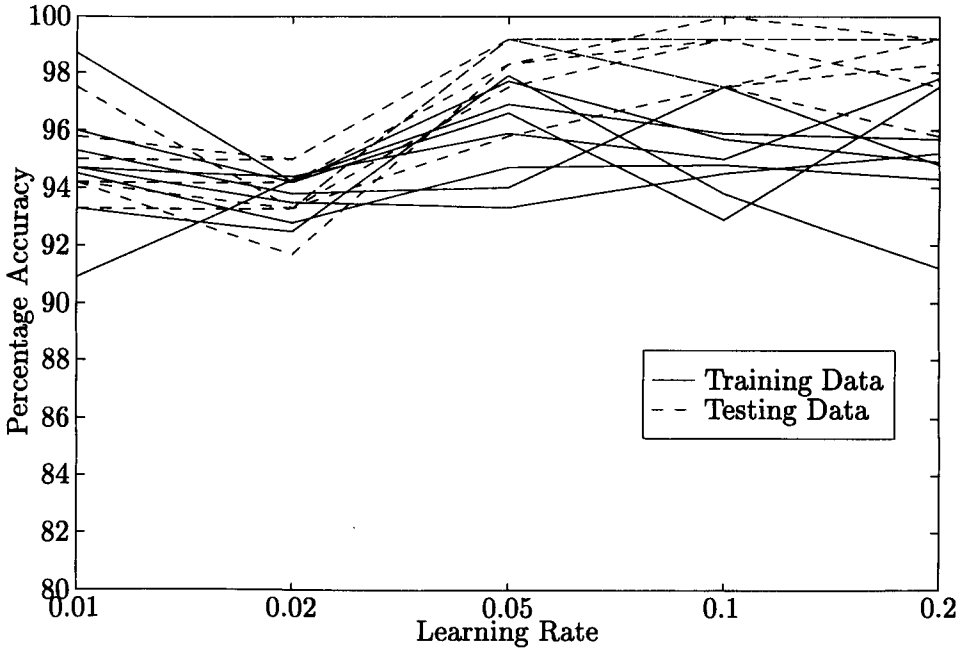


(b) Data set m04

Figure C.1 Moustache classification - comparison of different numbers of hidden neurons with constant data set and learning rate

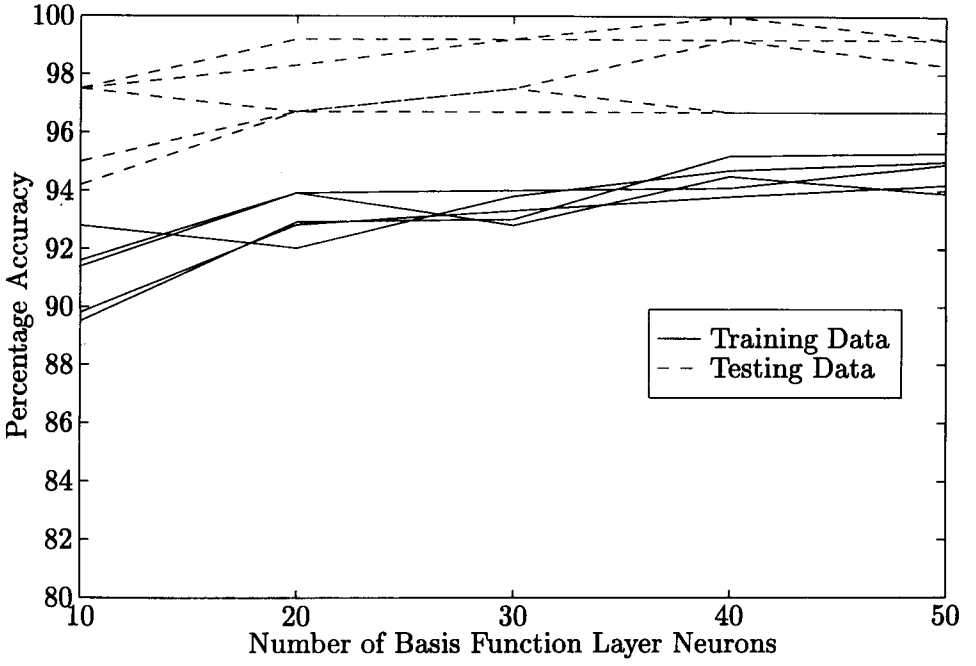


(a) Data Set m01

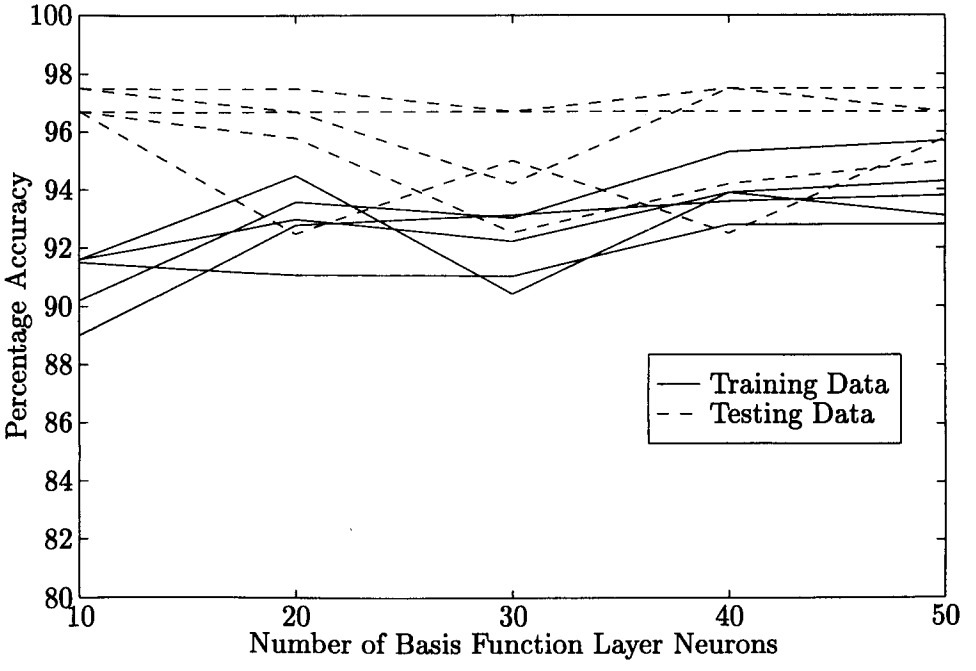


(b) Data Set m04

Figure C.2 Moustache classification - comparison of learning rates using a constant MLP topology and data set

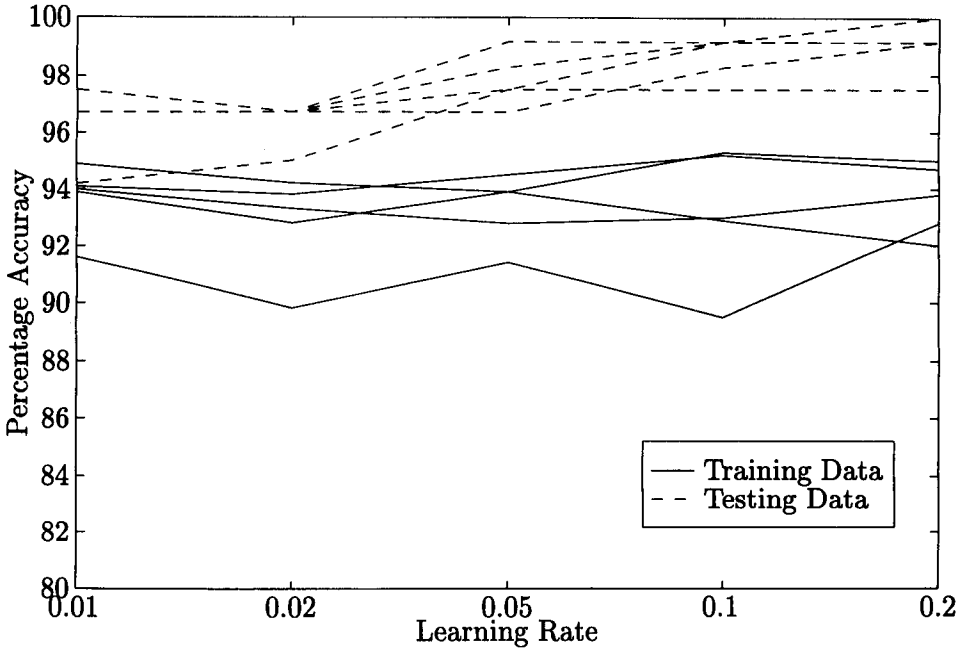


(a) Data set m01

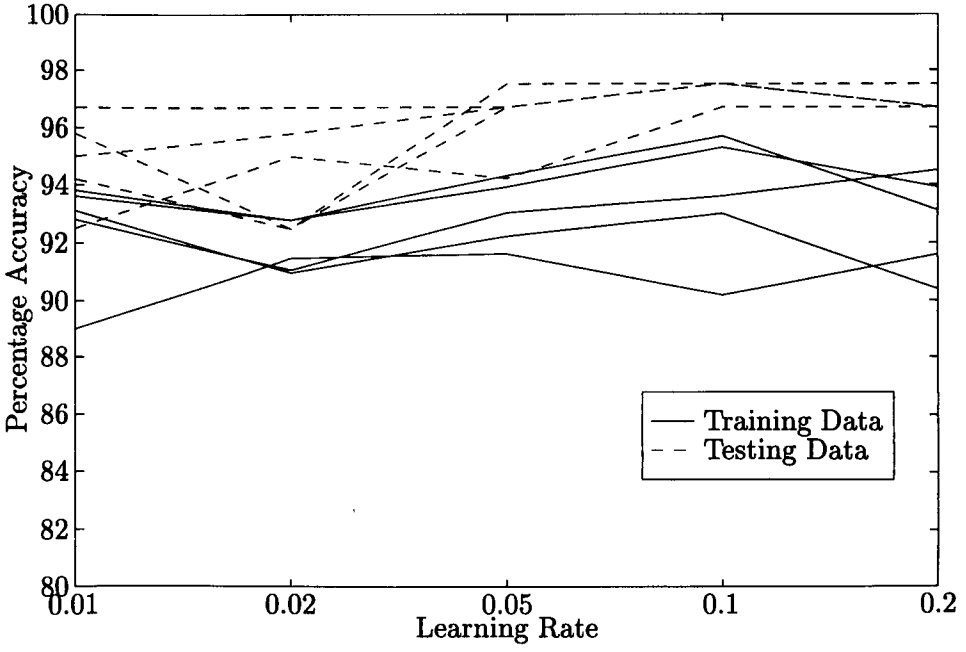


(b) Data set m04

Figure C.3 Moustache classification - comparison of different numbers of basis function layer neurons with constant data set and learning rate

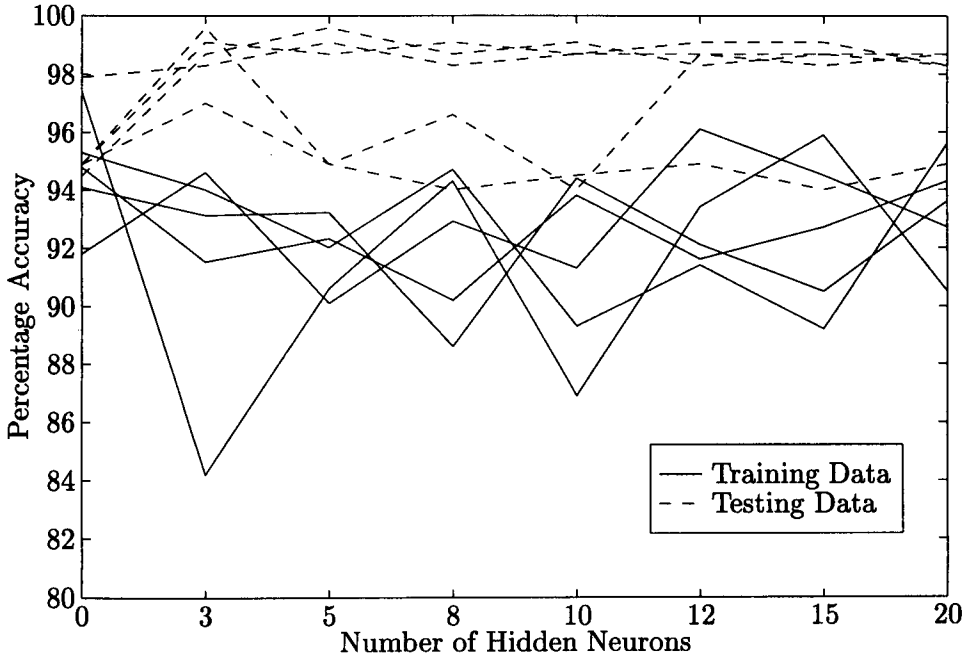


(a) Data Set m01

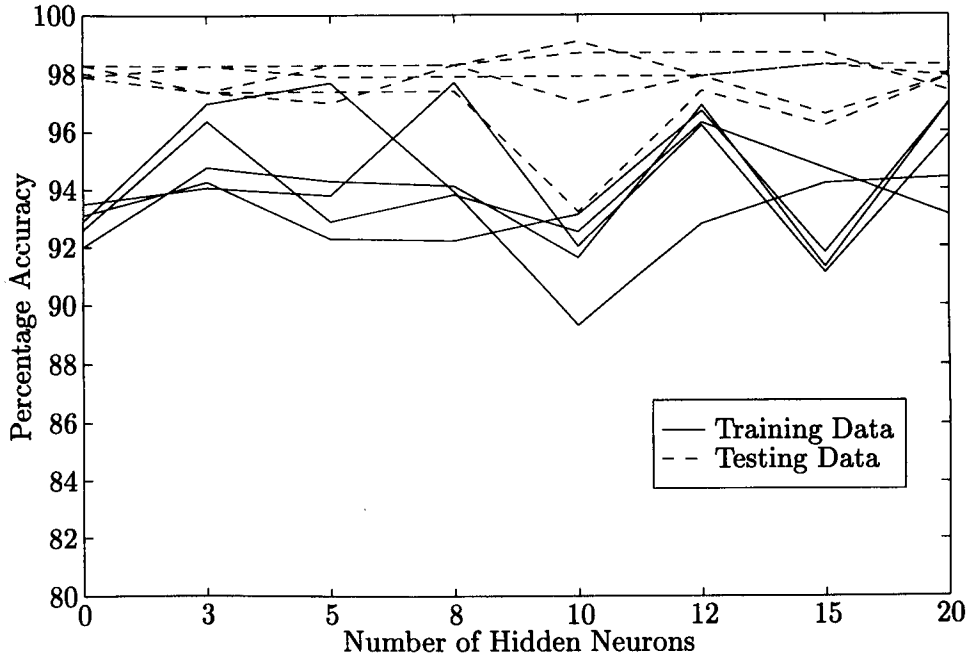


(b) Data Set m04

Figure C.4 Moustache classification - comparison of learning rates using a constant RBFN topology and data set

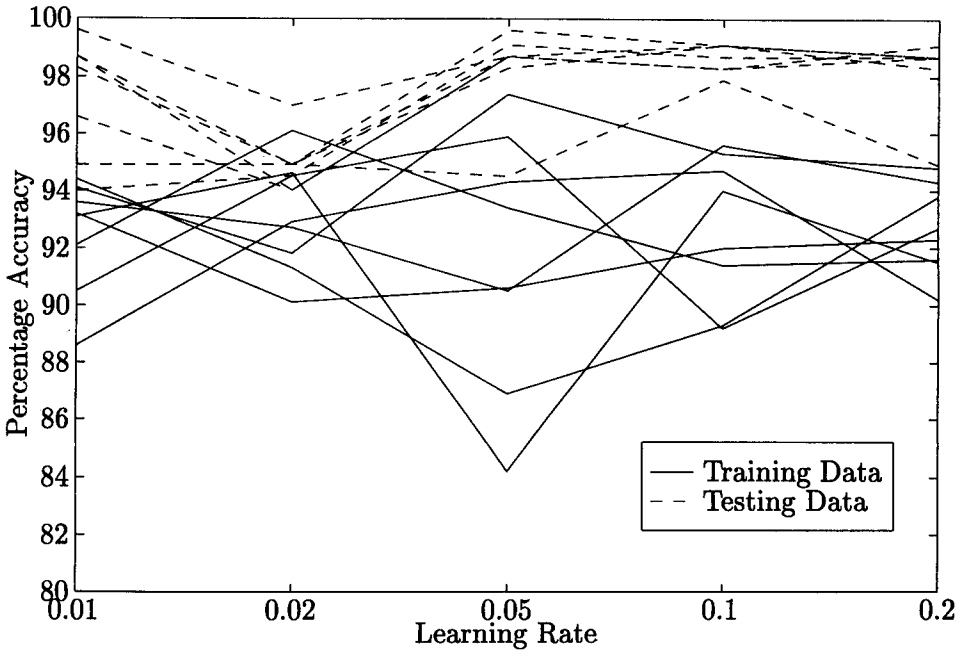


(a) Data set b02

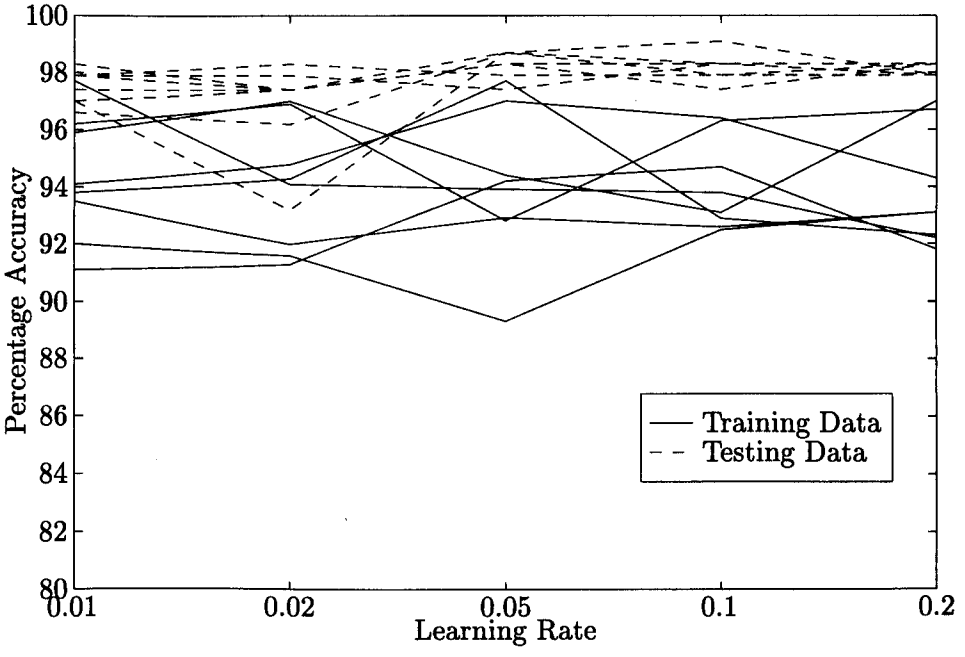


(b) Data set b04

Figure C.5 Beard classification - comparison of different numbers of hidden neurons with constant data set and learning rate

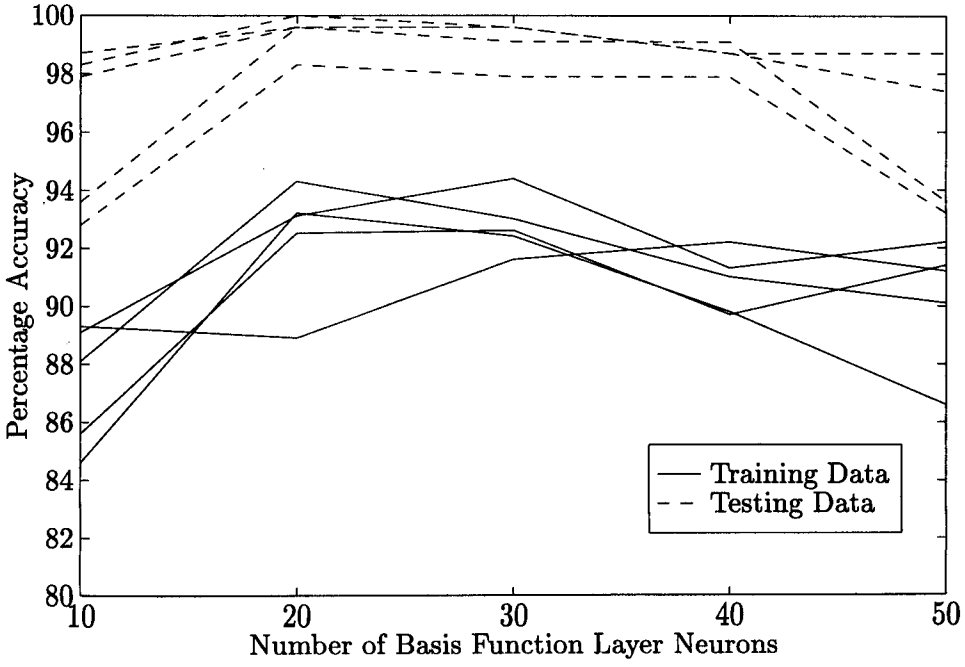


(a) Data Set b02

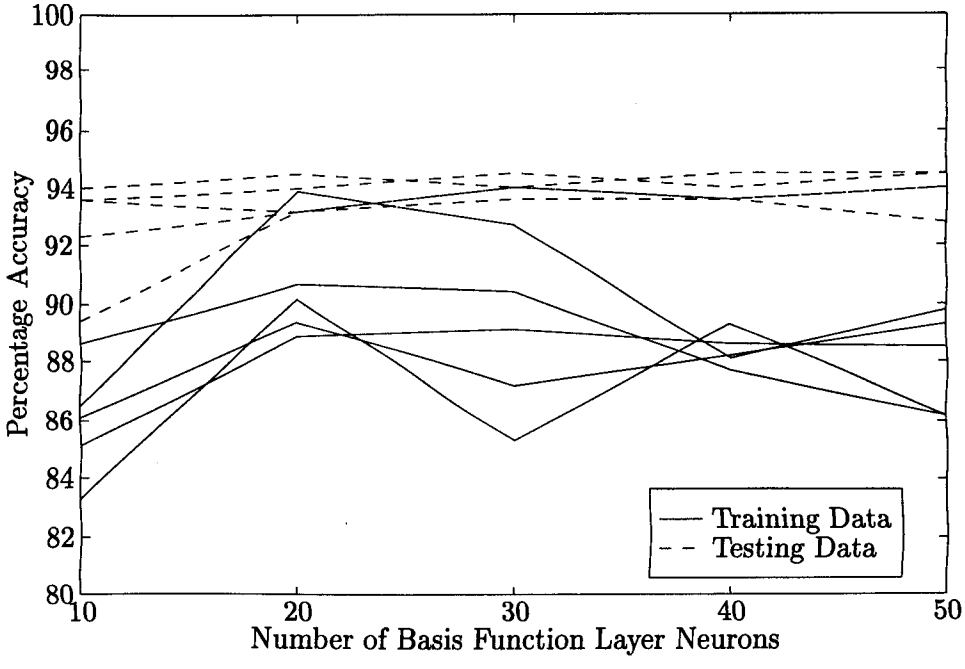


(b) Data Set b04

Figure C.6 Beard classification - comparison of learning rates using a constant MLP topology and data set

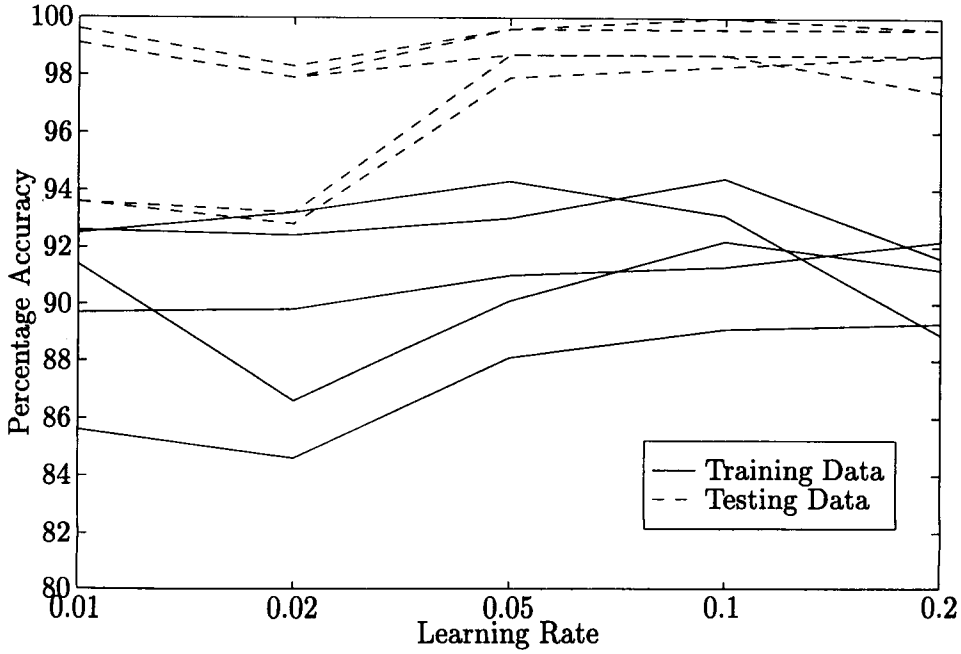


(a) Data set b02

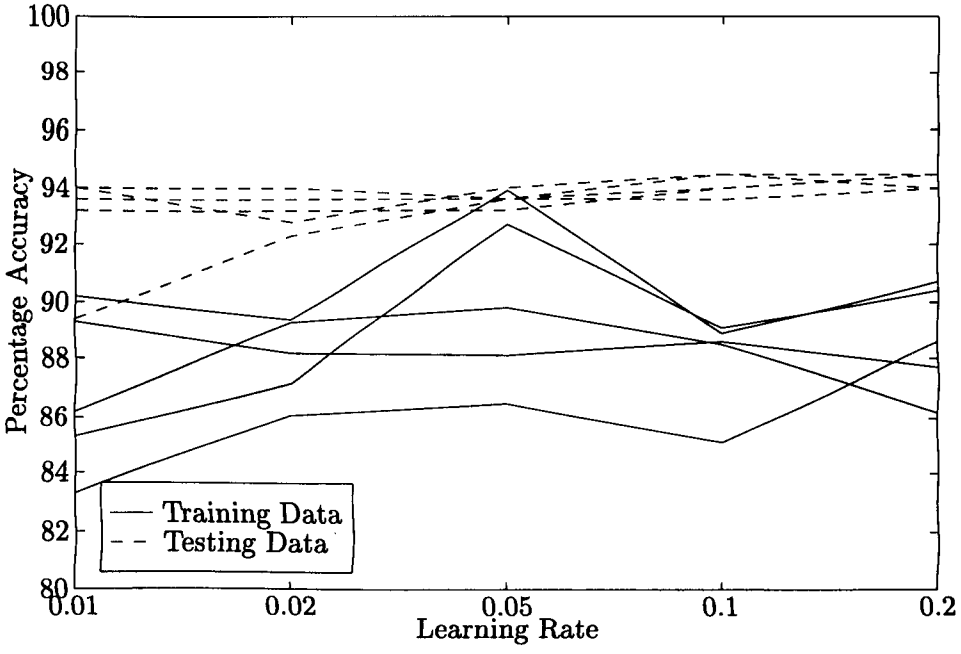


(b) Data set b04

Figure C.7 Beard classification - comparison of different numbers of basis function layer neurons with constant data set and learning rate

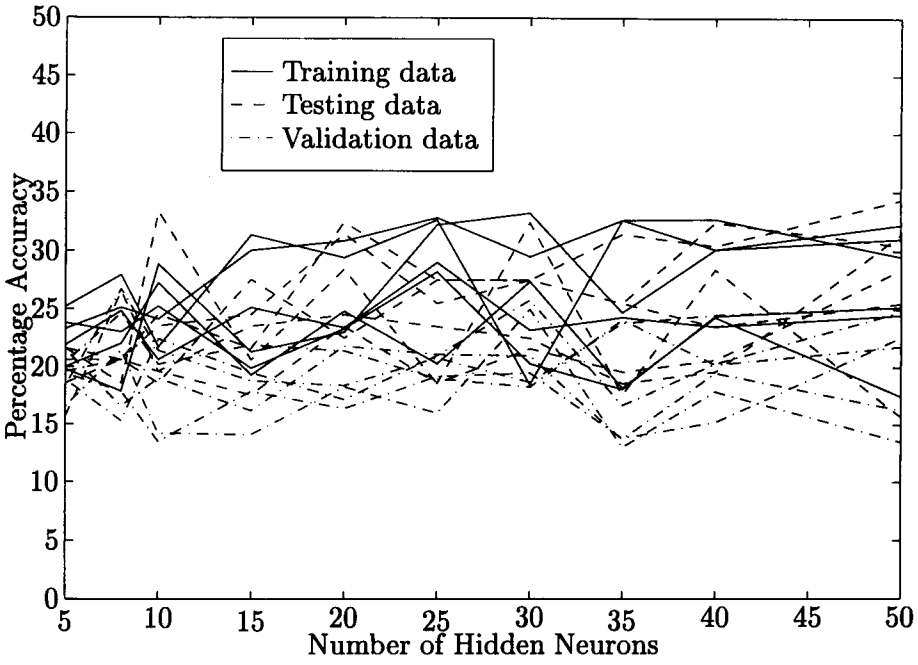


(a) Data Set b02

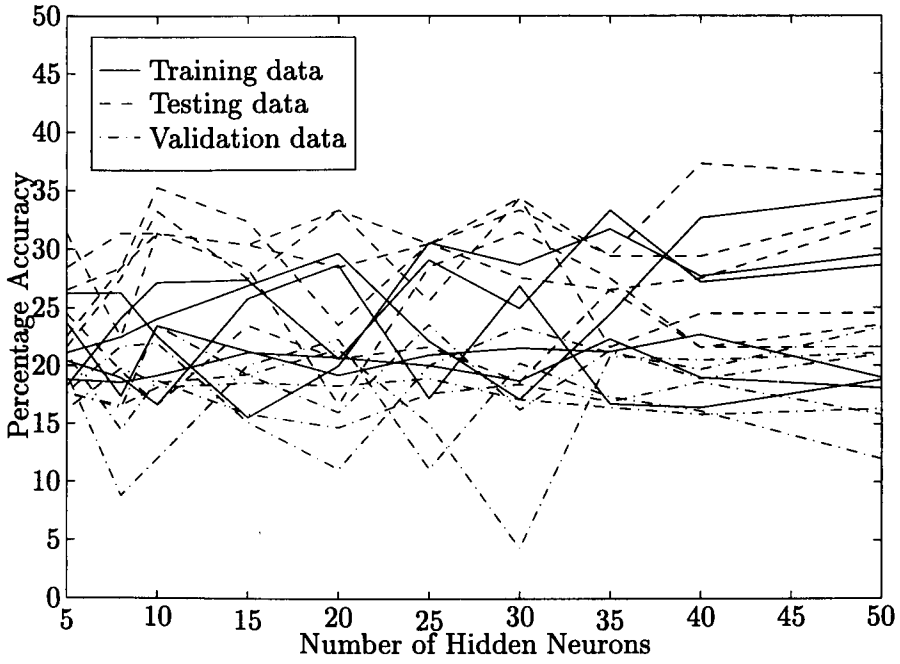


(b) Data Set b04

Figure C.8 Beard classification - comparison of learning rates using a constant RBFN topology and data set

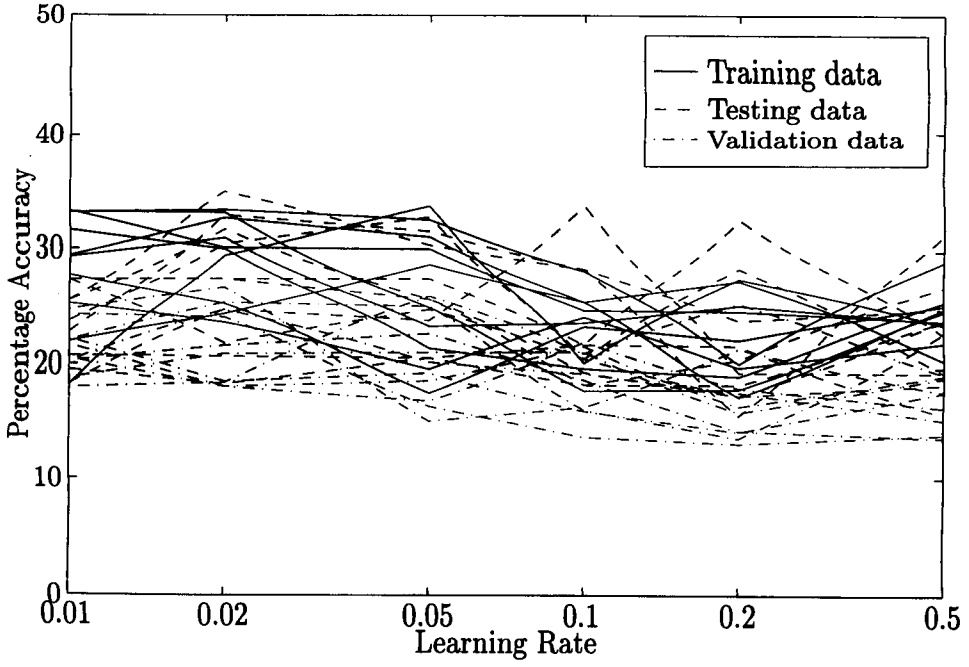


(a) Data set 02

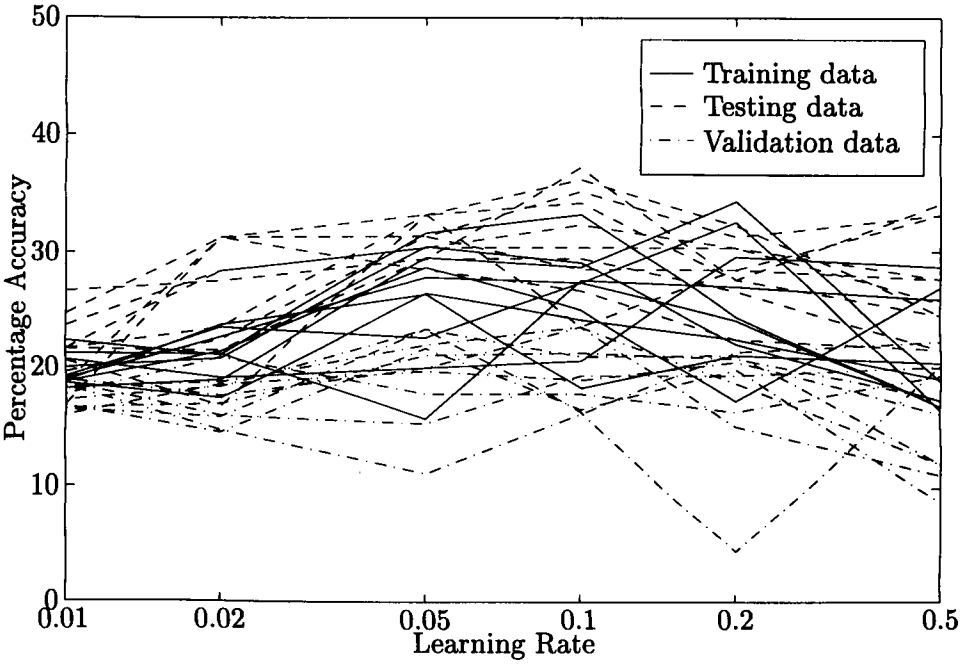


(b) Data set 09

Figure C.9 Eye colour classification - comparison of different numbers of hidden layer neurons in an MLP

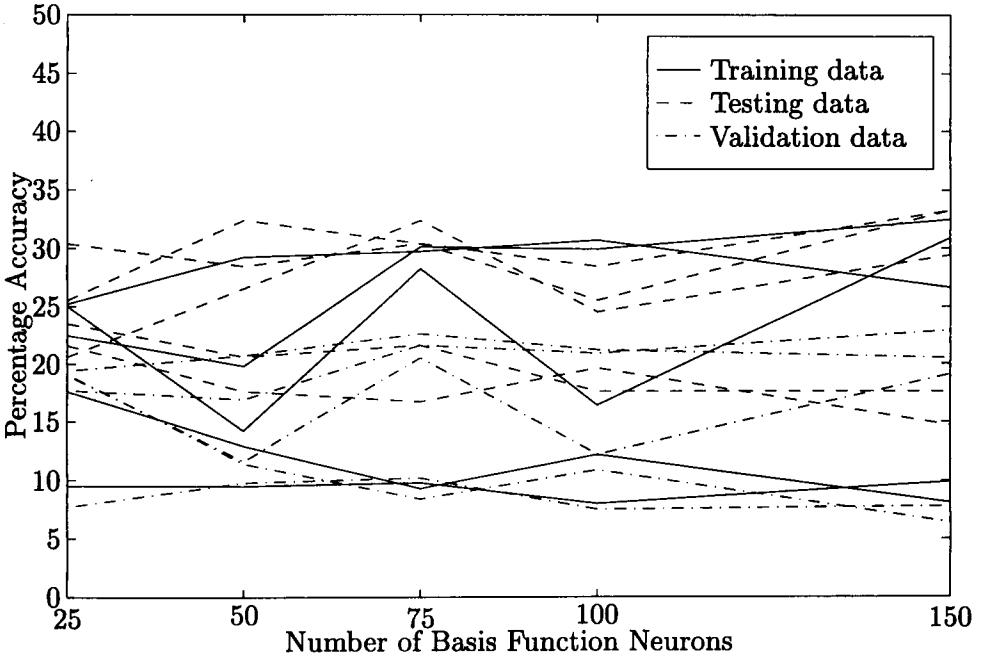


(a) Data set 02

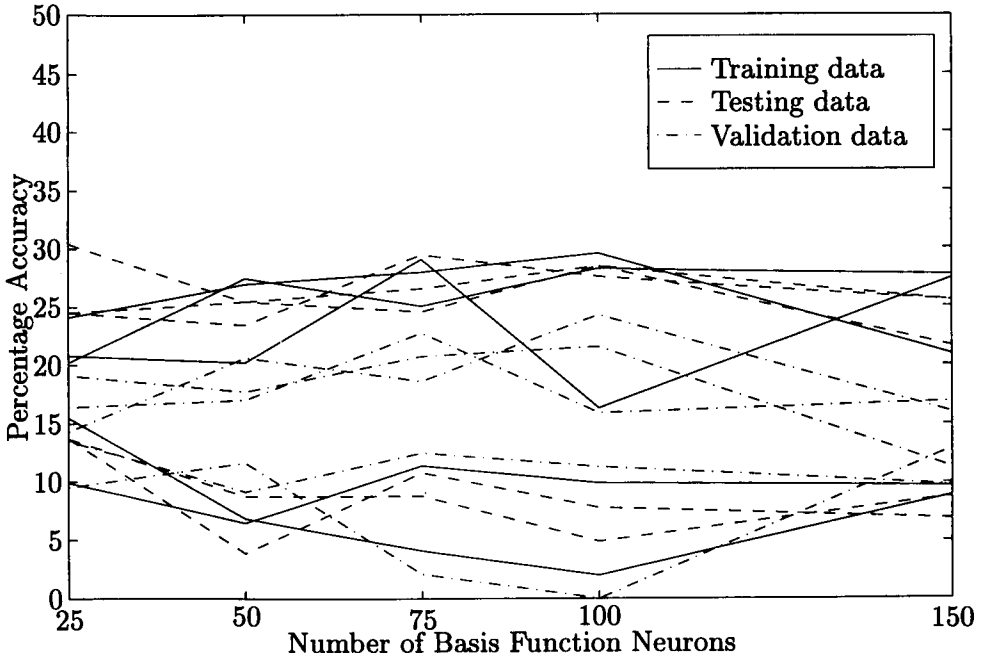


(b) Data set 09

Figure C.10 Eye colour classification - comparison of different learning rates in an MLP

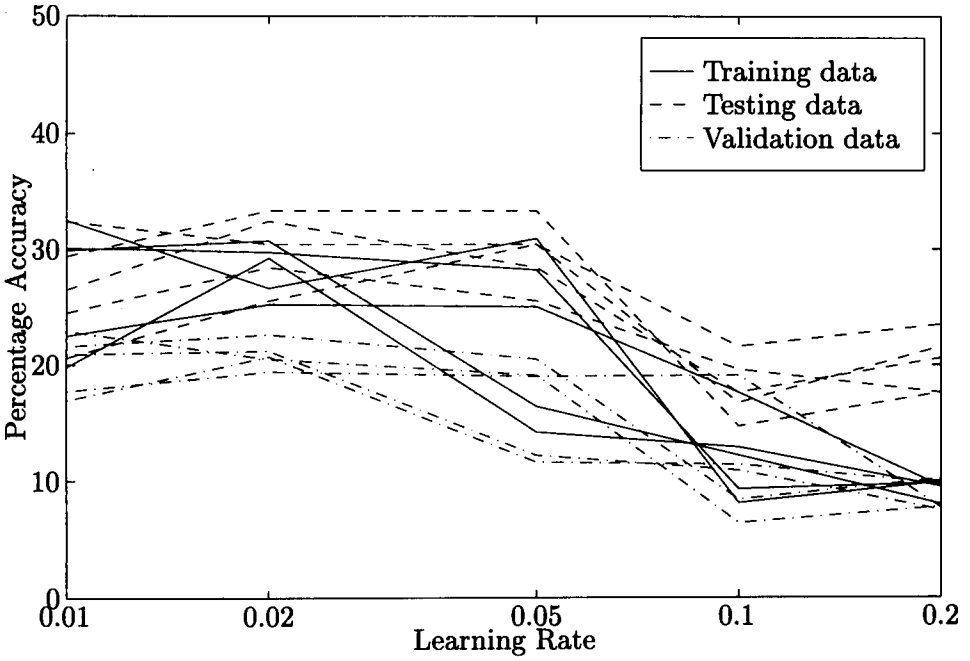


(a) Data set 02

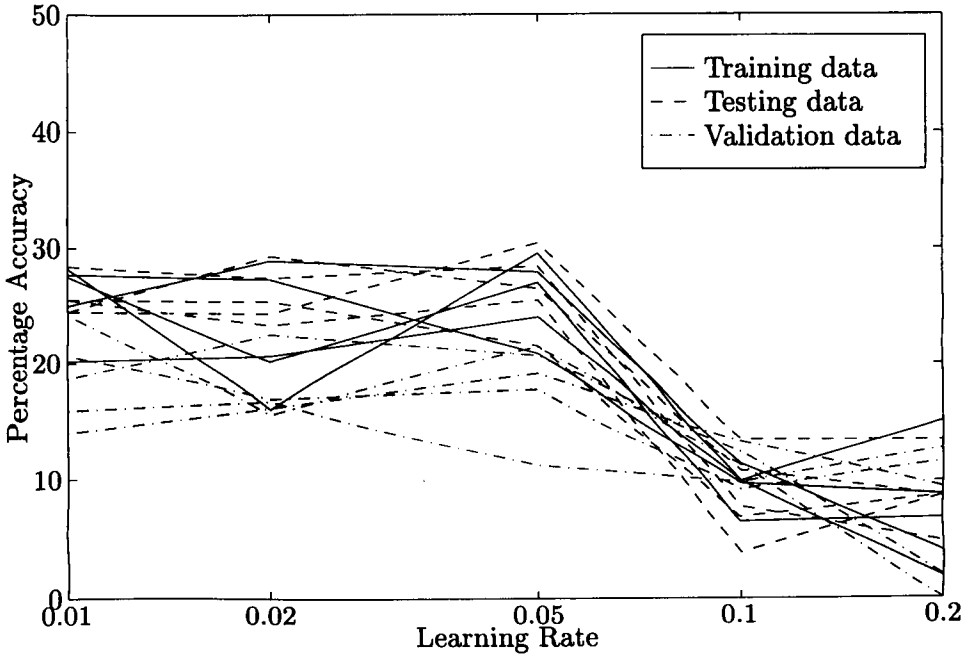


(b) Data set 09

Figure C.11 Eye colour classification - comparison of different numbers of basis function neurons in an RBFN

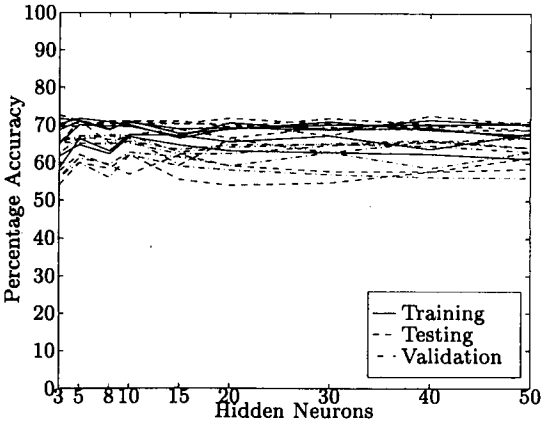


(a) Data set 02

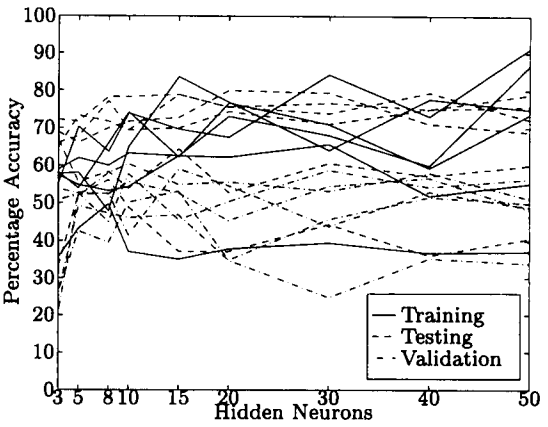


(b) Data set 09

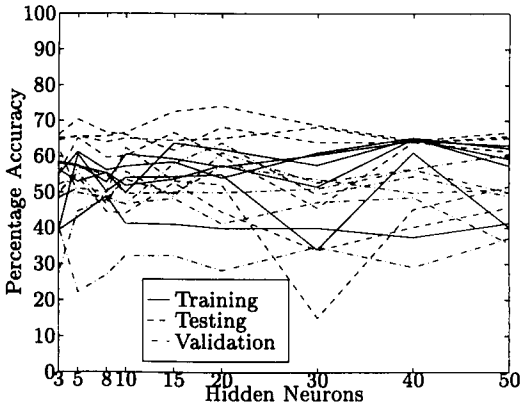
Figure C.12 Eye colour classification - comparison of different learning rates in an RBFN



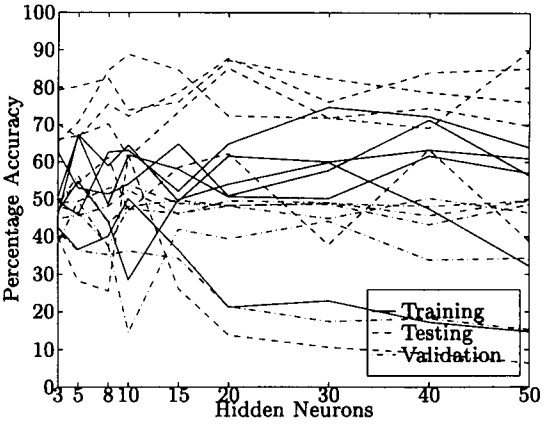
(a) Blue Eyes



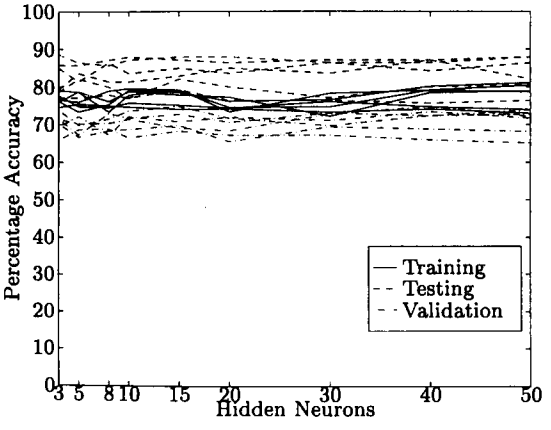
(b) Grey Eyes



(c) Green Eyes

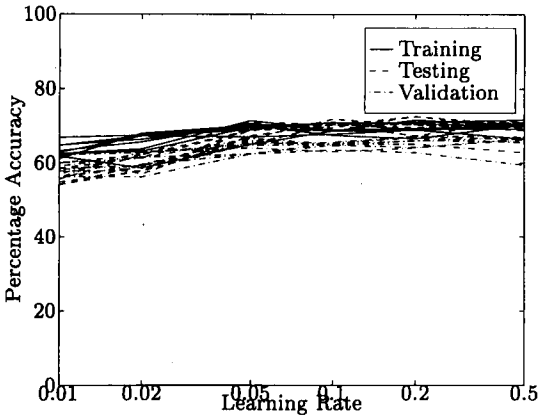


(d) Hazel Eyes

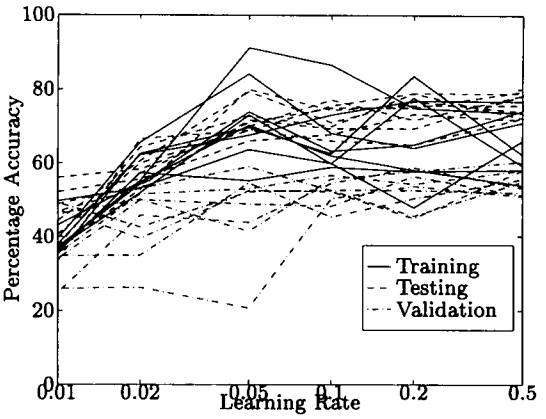


(e) Brown Eyes

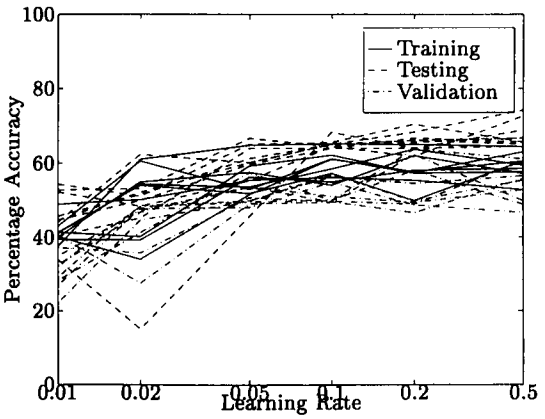
Figure C.13 Comparison of hidden layer size used in single eye colour classification with grey data sets



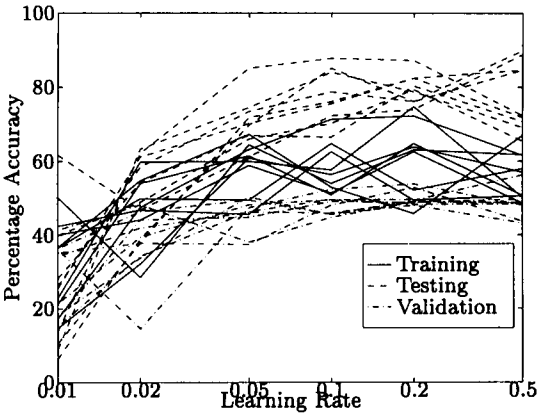
(a) Blue Eyes



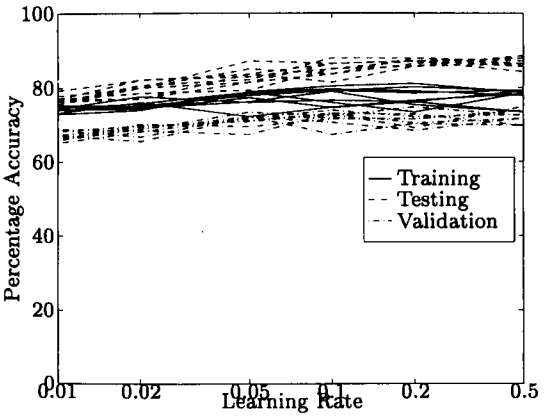
(b) Grey Eyes



(c) Green Eyes

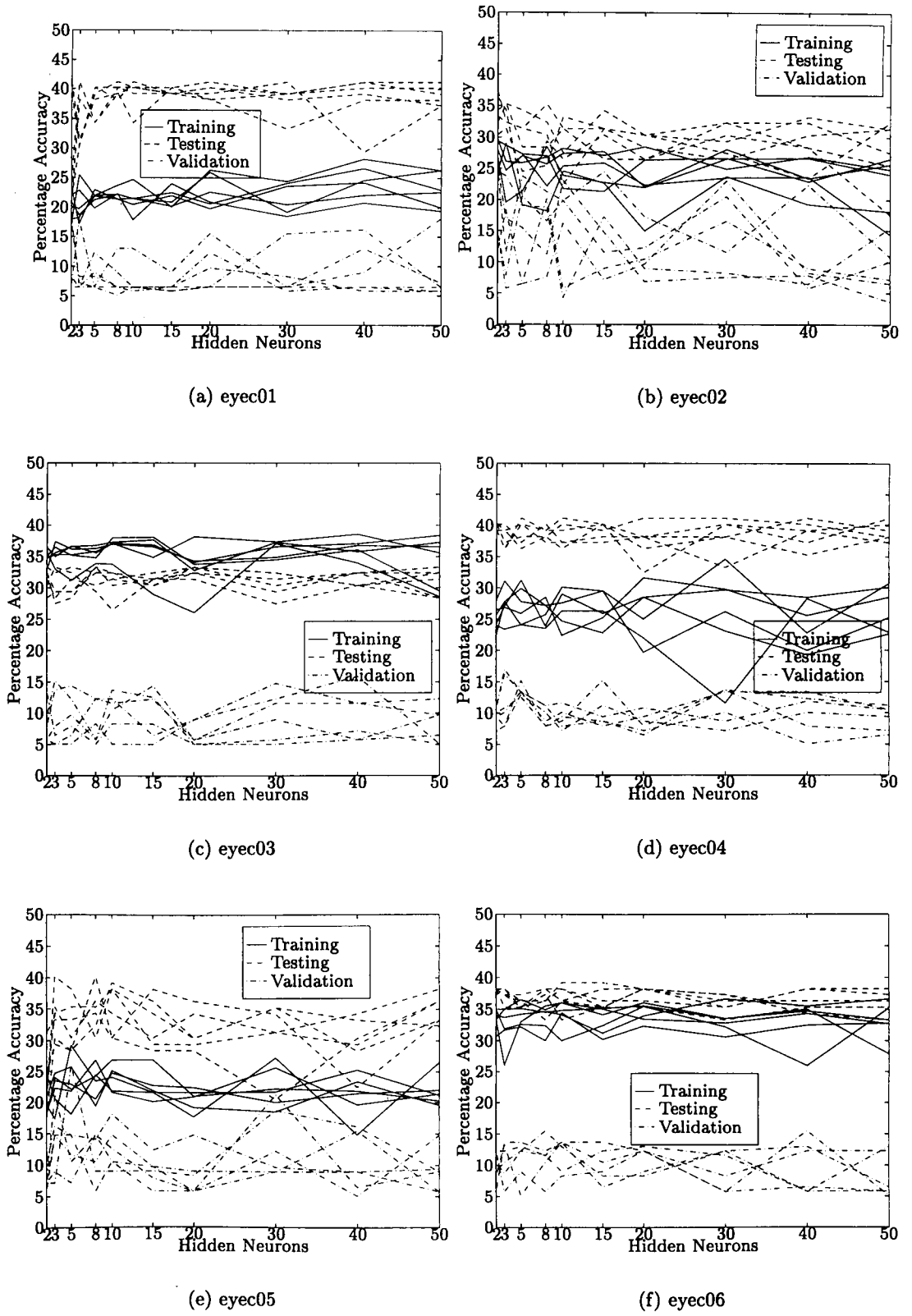


(d) Hazel Eyes



(e) Brown Eyes

Figure C.14 Comparison of learning rate used in single eye colour classification with grey data sets (MLP network)



(a) eyec01

(b) eyec02

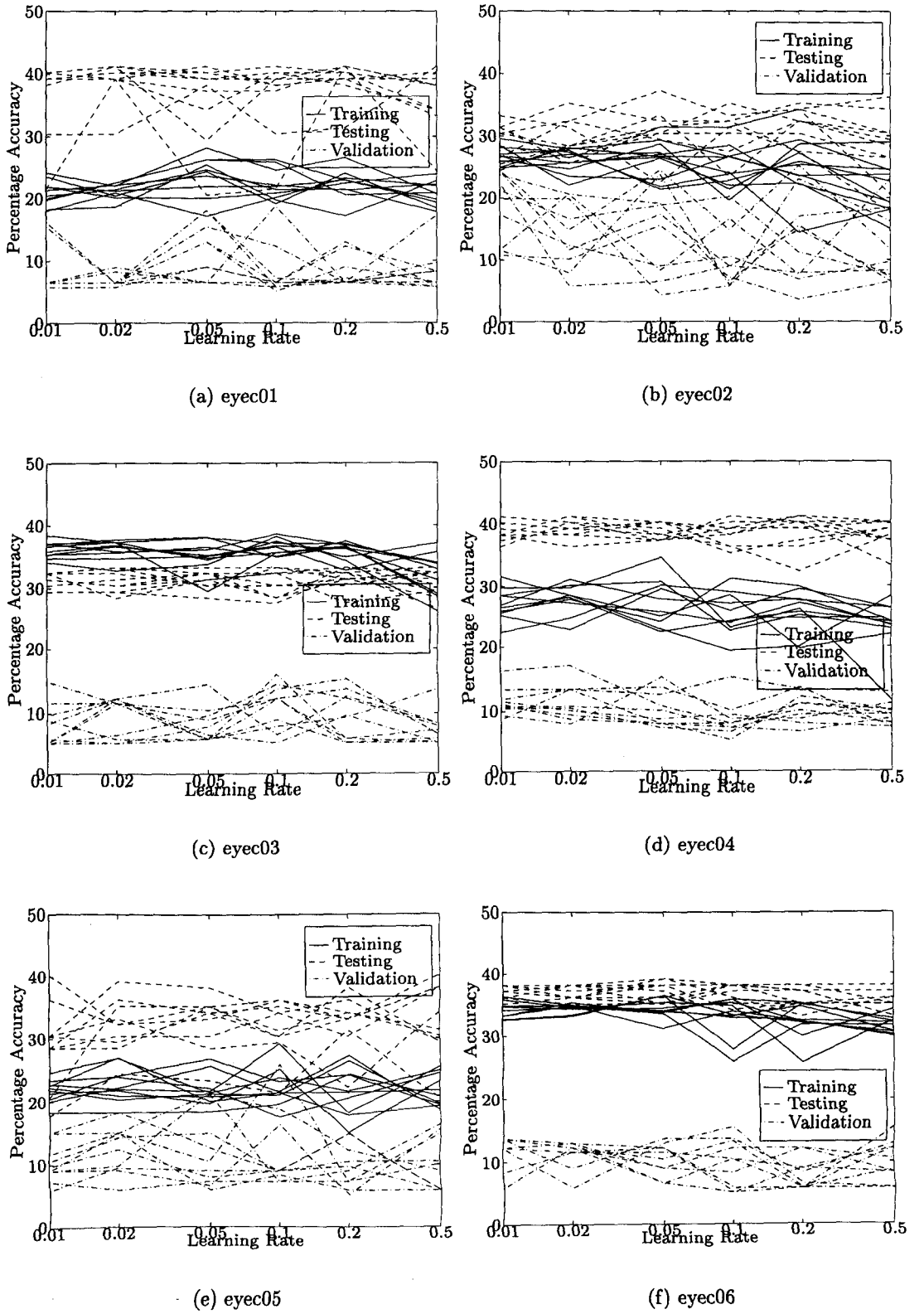
(c) eyec03

(d) eyec04

(e) eyec05

(f) eyec06

Figure C.15 Comparison of hidden layer size used in combining single eye colour classification networks (MLP network)



(a) eyec01

(b) eyec02

(c) eyec03

(d) eyec04

(e) eyec05

(f) eyec06

Figure C.16 Comparison of learning rate used in combining single eye colour classification networks (MLP network)

Appendix D

Publication from work in this thesis

The following paper was presented at Engineering Applications of Neural Networks 96 in London 17-19 June 1996. The work it is based on is related to that presented in chapters 5 and 6 of this thesis although the precise experiments are different. Following the work used for the paper, the experiments were re-run using NeuralWorks in order to allow for the use of the RBFN simulations.

Neural Networks for the Classification of Facial Features

N.D. Porter
Department of Engineering, University of Warwick, Coventry CV4 7AL, UK.

Abstract

Artificial neural networks (ANNS) are applied to the classification of facial features from digitised images of faces. From a set of 1000 faces, sections of images detailing specific features are extracted and presented to the ANNS which classify the features. The positions of the features in question are known and are used to locate the appropriate image section. Potential is shown for this technique with best results between 87% and 98% for the different features examined.

1. Introduction

Much work is being done on facial recognition in order to produce systems that identify people by recognising their faces [2, 6]. Many techniques are being used which can be broadly divided into two areas. Some locate feature points within the face image[9] and record these points while others match sections of the image with sections stored in the database of known faces [1].

Another aspect to facial image processing is that of producing a description of a face from the image. Describing a face is an imprecise task since there are no standard measures to use and, as the descriptions are usually given by humans, there will be much variation. For example what one person may describe as a long face, another may say is of average length. Some "descriptive" systems have been developed to reduce the dimensionality of the image data and they have resulted in measures which have little meaning to the human observer. These have frequently been some form of mathematical measure taken from the grey scale image such as the eigenface method used by Kirby and Sirovich [4] and the vector quantisation method used by Sutherland *et al* [8]

A set of feature measures developed at Aberdeen University[7] describe the face using human understandable measures such as *face length* or *colour of eyes*. This paper describes work concerned with automating the extraction of these descriptive measures using ANNS.

2. Experimental data

The data used here consists of a set of 1000 faces each digitised from photographs to 24 bit colour at a resolution of 384×512 pixels. This set was originally prepared by Aberdeen University and has recently been re-digitised by the Home Office for their face processing research. Fifty "natural" descriptive measures are associated with each of the images.

The data was first used in research into producing a descriptive index for a set of face images using the "natural" descriptive. The faces were presented to a set of jurors who gave their judgements on the measures in question. In addition, the positions of a set of 37 feature points were measured on the photographs, the data was stored in a computer and from these measures were taken which were correlated with the human estimated descriptors. Those human measures that matched closely were then replaced by the mathematical ones. For work on automatically extracting some of the physical features see [3].

While some of the measures can be mathematically derived from the positions of feature points, this work investigates the measurement of features that do not fall into this category using ANNS.

3. Experimental Methods and Results

The experiments were performed using a standard multi-layer perceptron trained using the back-propagation technique. Different network topologies were tried to obtain the optimum solution. The networks were trained using the Aspirin simulator package on Sun SPARC workstations.

Training data was taken as areas extracted from the original images, where the feature points were used to locate the areas of interest. While the images were supplied in 24 bit colour, initial work used grey scale images to reduce the complexity of the system. There is no set equation for this conversion, however, the one used (a weighted sum of the red, green and blue values) [5, p. 329], gives a good contrast over the final image.

3.1. Simple features - moustache and beard

The first features examined were the moustache and beard which served as a test for the feasibility of further development. To fully cover the area taken up by a moustache at the image resolution would require 100×25 pixels, giving 2500 inputs to the network for a grey scale image. This many inputs could cause problems with long training times and, with the limited quantity of data, generalisation is likely to be poor. Hence, the image areas extracted training were sub-sampled by pixel averaging in the horizontal and vertical directions to reduce the number of inputs to the network. Different levels of sub-sampling over various image sizes were tried resulting in input sizes ranging from 100 to 20 as shown in Table 1.

Table: 1: Classification accuracy of moustache data using two output neurons

Image size	Sub-sample	No. net. i/p	Training acc.	Testing acc.
100×25	5×5	100	99.0%	99.5%
100×5	5×5	20	95.6%	98.4%
100×25	25×5	20	97.4%	99.5%
100×25	5×25	20	95.4%	98.9%

The networks used had two outputs and the number of hidden neurons was varied between 5 and 15 to identify the network with the best generalisation. All the data sets were centred round a point 12 pixels above the middle of the top lip. The networks were trained with data from 900 images, saving the weights regularly as the training progressed. The saved networks were then tested by presentation of both the training and testing data, which was taken from the remaining 100 images, to evaluate the classification rates achieved. These were examined to find the set of weights that produced the highest testing accuracy which was taken as the best network solution. Long training cycles of 100000 itterations were used to ensure that the best solution was found.

Table 1 shows results obtained for identification of moustaches with some of the data sets used. The two outputs were arranged with one indicating the presence and one indicating the absence of the moustache. The output neurons gave values in the range $[(-1),1]$ and a threshold[1] of 0.2 was applied to the output such that outputs less than this were unclassified. The accuracy measure in the tables of results is the sum of the correct classifications over the number of trials.

Another scheme was investigated with a single output from the network where an output of 1 indicated the presence and -1 indicated the absence of a moustache. Again the number of hidden neurons was varied and a threshold was applied such that outputs with a magnitude less than 0.2 were said to be unclassified resulting in similar accuracies though with shorter training cycles.

The same techniques were applied to the identification of beards with a maximum accuracy of 94% achieved on testing. The data sets for these experiments were centred around a point either 35 or 50 pixels below the middle of the bottom lip depending on the size of image extracted. The images ranged in size from 200×70 to 50×40; best results being achieved with an image 100×70.

In the cases of both the beards and moustaches, there were many more examples in the set of face images without these features than there were with. If the networks had been trained using the data sets with the different classes represented in these proportions then it is most likely that they would have learnt only the cases of faces without the feature that was to be identified as this would have been the strongest influence in the training data leading to all faces being classified into this group. To compensate for this the faces with the feature were presented to the network repeatedly so as to balance the numbers in each class. This method of balancing the training sets has the advantage of not throwing away any valuable training data while overcoming the problem of an out of balance data set and this was the chosen method.

3.2. Complex feature - eye colour

The classifications discussed so far required outputs that indicated the presence or absence of a feature while other, more complex features require a grading as the output. In the encodings used

[1] The use of a threshold reduces the effect that noise can have in altering the result of the classification.

by Aberdeen, eye colour is categorised into one of five colours, (blue, grey, green, hazel and brown) so the networks used for this had five outputs, one for each colour. The correct result was assumed to be the output with the highest value provided that that output was at least 0.2 greater than the next largest. If no output met this criteria, then the result was deemed to be unclassified.

As before, different sizes of image were taken with respect to a face point; in this case the centre of an eye. Most training data sets were centred around this point but one was taken offset to one side to avoid the eye pupil which is irrelevant to the colour measures and also to avoid areas of the eye that were prone to reflections from the lighting used in the photography.

Since the images were in colour, this provided opportunity to present data to neural networks using different colour representations. The original images were supplied in an RGB format representing the red, green and blue components of each pixel. Other representations can be derived from this including HSV where the values represent the hue, saturation and value or brightness of the colour. As HSV is designed to bear resemblance to human recognition of colours, it was thought that this representation would provide useful information for the classification.

Table: 2: Classification accuracy of eye colour - all colours

Image size	Colour	Sub-sample	No. net. i/p	Training acc.	Testing acc.
20 × 7	grey	1 × 1	140	65.6%	48.4%
20 × 8	grey	2 × 2	40	50.3%	49.0%
8 × 8	grey	1 × 1	64	59.1%	53.6%
30 × 20	grey	5 × 5	24	60.5%	51.0%
8 × 8	blue	2 × 2	16	48.2%	51.0%

In addition to grey-scale, hue information from the HSV scheme and each of the individual red, green and blue components were used in producing image segments for training. Table 2 shows the results obtained with some of the different settings. The best results from these experiments were obtained using an 8 × 8 image section offset five pixels from the centre of the eye.

From visual inspection of the images, it was apparent that some of the classes are more easily recognisable than others so investigation was made into training a series of networks to recognise just a single colour of eye rather than a single network for all colours. Networks were trained using the 8 × 8 section of image in different colour representations to revealed which colours of eye were easiest to detect and which colour representation best distinguished any given eye colour. The training data was sub-sampled to reduce the image to 16 data points, requiring networks with 16 inputs and the number of hidden neurons was varied once more. The best results obtained for each eye colour are given in Table 3. The same threshold system as for the single output moustache identifier was used.

Table: 3: Classification accuracy of eye colour - single colours

Eye colour	Data colour	No. Hidden neurons	Training acc.	Testing acc.
Blue	blue	4	82.8%	82.2%
Grey	green	5	82.8%	79.4%
Green	hue	5	83.5%	64.5%
Hazel	blue	4	72.5%	78.2%
Brown	green	5	84.0%	87.1%

4. Discussion and Conclusions

A number of different points may be drawn from the results of these experiments.

When the moustache identifier was changed from two outputs to one output, the best solution was achieved in a lower number of training epochs. This may be explained by the reduced number of weights between the hidden and output layers giving a reduction in the dimensionality of the weight space that is searched for the solution.

Examining the images of those that were misclassified, it was apparent that the moustaches that were missed were often very small and involved light coloured hair making them difficult to see even in the full colour images. As for those without moustaches that the networks said had them, one may have expected these to be people with darker skin confusing the network. This could not generally be said to be the case as these misclassifications involved all different kinds of faces. It was noted that while a couple of faces appear regularly as misclassified during the different tests, generally each network produced a different set of faces that it failed to classify.

The classification of eye colour raises the question of the best data representation to be used to extract colour information. It was thought that the use of the hue component from the HSV representation would give useful colour information, however when all five classes were to be identified by a single network, the grey scale representation came out consistently better. Using separate networks for each of the classes of eye colour, allowed us to use different data representations for each class and as Table 3 shows it was not always the representation that might be expected that performed the best. For example, the class of green eyes was best found using the hue data set though the green was the set that performed the worst. Visual inspection of the data set showed that the blue and brown were the most clearly distinguishable classes while there could easily be some confusion between the members of the other three and this is born out by the fact that the accuracies achieved for these classes were lower than those for the blue and brown. As human descriptions of colour are imprecise, perfect response of a system to match these measurements is not possible. However the basic categorisation of the colours about which there is little dispute is achievable and this has been demonstrated with the networks used here.

This work has shown that a MLP based system is capable of being used for facial feature classification purposes. The identification of simple features such as beards and moustaches was straight forward with around 98% accuracy being achieved.

The more complex feature of eye colour was classified with accuracies between 64% and 87% when individual networks were used in the identification of each class. Further work now needs to be done on combining these individual networks to produce a single system for eye colour detection.

Acknowledgements

The author wishes to thank EPSRC and the UK Home Office who have supported this work and the Department of Psychology at Aberdeen University for their contribution to the provision of the database used.

References

[1] H. M. Bouattour, F. F. Soulié, and E. Viennet. Neural nets for human face recognition. *IEEE*, pages III–700–III–704, 1992.

[2] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, Oct. 1993.

[3] I. Craw, H. Ellis, and J. R. Lishman. Automatic extraction of face-features. *Pattern Recognition Letters*, 5(2):183–187, Feb. 1987.

[4] M. Kirby and L. Sirovich. Application of the Karhunen–Loève procedure for the characterisation of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, Jan. 1990.

[5] S. Rimmer. *Bit-mapped graphics.* McGraw-Hill, 1993.

[6] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[7] J. W. Shepherd. An interactive computer system for retreiving faces. In *Aspects of Face Processing*, pages 398–409. NATO, July 1985.

[8] K. Sutherland, D. Renshaw, and P. B. Denyer. Automatic face recognition. In *First International Conference on Intelligent Systems Engineering*, pages 29–34. IEE, Aug. 1992.

[9] C. J. Wu and J. S. Huang. Human face profile recognition by computer. *Pattern Recognition*, 23:255–260, 1990.

References

[1] Specification for the development of a facial recognition system, September 1987. Home Office ref C4971/SS.

[2] R. Anand, K.G. Mehrotha, C.K. Mohan, and Ranka S. An improved algorithm for neural network classification of imbalances training sets. *IEEE Transactions on Neural Networks*, 4(6):962–969, November 1993.

[3] R. J. Baron. Mechanisms of human facial recognition. *International Journal of Man-Machine Studies*, 15:137–178, 1981.

[4] D. Beymer and T. Poggio. Face recognition from one example. A.I. Memo No. 1536, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, September 1995.

[5] H. M. Bouattour, F. F. Soulié, and E. Viennet. Neural nets for human face recognition. *IEEE*, pages III–700–III–704, 1992.

[6] H. M. Bouattour, F. F. Soulié, and E. Viennet. Solving the human face recognition task using neural nets. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks*, volume 2, pages 1595–1598. Elsevier Science, 1992.

[7] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, October 1993.

[8] T. Brunelli and T. Poggio. Face recognition through geometrical features. In *Proceedings of ECCV '92*, pages 792–800, S. Margherita Ligure, 1992.

[9] A. M. Burton and V. Bruce. An interactive activation model of human face recognition. In *Neural Computing Research and Applications*, pages 73–83, Queen's University of Belfast, Northern Ireland, June 1992.

[10] Rama Chellappa, Charles L. Wilson, and Saad Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–740, 1995.

[11] C. Chen and S.-P. Chiang. Detection of human faces in colour images. *IEE Proceedings. Vision, Image and Signal Processing*, 144(6):384–388, December 1997.

[12] T. Cox. The fastphoto field trial. Technical report, Hertfordshire Constabulary, 1988.

[13] I. Craw, H. Ellis, and J. R. Lishman. Automatic extraction of face-features. *Pattern Recognition Letters*, 5(2):183–187, February 1987.

[14] G. M. Davies. Face recall systems. In G. M. Davies, H. D. Ellis, and J. W. Shepherd, editors, *Perceiving and Remebering Faces*, Cognition and Perception, chapter 10, pages 227–250. Academic Press, 1981.

[15] R. M. Debenham and S. C. J. Garth. The detection of eyes in facial images using radial basis functions. In R. Linggard, D. J. Myers, and C. Nightingale, editors, *Neural Networks for Vision, Speach and Natural Language*, volume 1 of *BT Telecommunications Series*, chapter 2, pages 50–64. Chapman Hall, 1992.

[16] S. Edelman, D. Reisfeld, and Y. Yeshurun. Learning to recognise faces from examples. *Lecture Notes in Computer Science*, pages 787–791, 1992.

[17] H. D. Ellis, G. M. Davies, and J. W. Shepherd. A critical examination of the Photofit system for recalling faces. *Ergonomics*, 21(4):297–307, 1978.

[18] H. D. Ellis, J. W. Shepherd, and G. M. Davies. An investigation of the use of the Photo-fit technique for recalling faces. *British Journal of Psychology*, 66(1):29–37, 1975.

[19] J. D. Foley, A. van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips. *Introduction to Computer Graphics*. Addison Wesley, 1994.

[20] K. Fukushima, S. Miyake, and T. Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(5):826–834, September 1983.

[21] R. G. Gallager. *Information Theory and Reliable Communication.* John Wiley & Sons, New York, 1968.

[22] A. G. Goldstein and J. Chance. Measuring psychological similarity of faces. *Bulletin of the Psychonomic Society*, 7:407–408, 1976.

[23] A. G. Goldstein and J. Chance. Judging face similarity in own and other races. *Journal of Psychology*, 98:185–193, 1978.

[24] A. J. Goldstein, L. D. Harmon, and A. B. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, May 1971.

[25] R. C. Gonzalez and R. C. Woods. *Digital Image Processing.* Addison-Wesley, 3rd edition edition, 1992.

[26] V. Govindaraju, D. B. Sher, R. K. Srihari, and S. N. Srihari. Locating human faces in newspaper photographs. In *Proceedings of the IEEE Computing Society Conference on Computer Vision and Pattern Recognition*, pages 549–554, 1989.

[27] K. Gurney. *An Introduction to Neural Networks.* Morgan Kaufmann, 1997.

[28] P. J. B. Hancock. Data representation in neural nets: An empirical study. In *Procedings of the 1988 Connectionist Models Summer School*, pages 11–20, September 1989.

[29] L. D. Harmon. The recognition of faces. *Scientific American*, 229(5):71–82, November 1973.

[30] R. Herpers, L. Witta, J. Bruske, and G. Sommer. Evaluation of local images structures applying a dcs network. In A. B. Bulsari, S. Kallio, and D. Tsaptsinos, editors, *Solving Engineering Problems with Neural Networks*, pages 305–312, London, June 1996.

[31] J. H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975. (2nd edition 1992).

[32] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences, USA*, 79:2554–2558, 1982.

[33] A. J. Howell and H. Buxton. Invariance in radial basis function neural networks in human face classification. Technical Report CSRP 365, University of Sussex at Brighton, February 1995.

[34] A.J. Howell and H. Buxton. Improving generalisation in radial basis function networks for face recognition. In A. B. Bulsari, S. Kallio, and D. Tsaptsinos, editors, *Solving Engineering Problems with Neural Networks*, pages 297–304, London, June 1996.

[35] M. K. Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8:179–187, 1962.

[36] C. L. Huang and C. W. Chen. Human facial feature extraction for face interpretation and recognition. *Pattern Recognition*, 25(12):1435–1444, 1992.

[37] Q. Jiang. Principal component analysis and neural network based face recognition. Ms thesis, Department of Computer Science, University of Chicago, 1996.

[38] I. T. Jolliffe. *Principle Components Analysis.* Springer-Verlag, 1986.

[39] M. Kirby and L. Sirovich. Application of the Karhunen–Loève procedure for the characterisation of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, January 1990.

[40] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 3 edition, 1989.

[41] B. Kumar, D. Casasent, and H. Murakami. Principle component imagery for statistical pattern recognition correlators. *Optical Engineering*, 21(1):43–47, 1982.

[42] M. Lando and S. Edelman. Generalization from a single view in face recognition. WISDOM Technical Reports in Computer Science CS95-02, Weizmann Institute of Science, 1995.

[43] K. R. Laughery, P. K. Fessher, D. R. Lenorovitz, and D. A. Yoblick. Time delay and similarity effects in facial recognition. *Journal of Applied Psychology*, 54(4):490–496, 1974.

[44] K. R. Laughery and R. H. Fowler. Sketch artist and Identi-Kit procedures for recalling faces. *Journal of Applied Psychology*, 65(3):307–316, 1980.

[45] K. R. Laughery, B. T. Rhodes, Jr., and G. W. Batten, Jr. Computer-guided recognition and retrieval of facial images. In G. M. Davies, H. D. Ellis, and J. W. Shepherd, editors, *Perceiving and Remebering Faces*, Cognition and Perception, chapter 11, pages 251–269. Academic Press, 1981.

[46] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, 4(2):4–22, April 1987.

[47] B. S. Manjunath, R. Chellappa, and C. von der Malsburg. A feature based approach to face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition '92*, pages 373–378, Champaign, IL., June 1992.

[48] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(115), 1943.

[49] E. Micheli-Tzanakou. What artificial neural networks can do. *Biomedical Engineering Society Newsletter*, 19(3), 1995. available as http://www.mecca.org/BME/BMES/bme19-3a.html.

[50] E. Micheli-Tzanakou and G. M. Binge. F-CORE:a fourier based image compression and reconstruction technique. *SPIE Proceedings on Visual Communication and Image Processing IV*, 1199:1563–1574, 1989.

[51] E. Micheli-Tzanakou and E. Harth. Determination of visual receptive fields by stochastic methods. *Biophysics Journal*, 15(42a), 1973.

[52] E. Micheli-Tzanakou, E. Uyeda, R. Ray, A. Sharma, R. Ramanujan, and J. Dong. Comparison of neural network algorithms for face recognition. *Simulation*, 64(1):37–50, 1995.

[53] M. L. Minsky and S. Papert. *Perceptrons.* MIT Press, Cambridge, MA and London, England, 1969. (2nd eddition 1989).

[54] B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE*, volume 2277. July 1994.

[55] B. Moghaddam and A. Pentland. An automatic system for model-based coding of faces. In *Proceedings of the IEEE Data Compression Conference*, Snowbird, Utah, March 1995.

[56] D. C. Montgomery and E. A. Peck. *Introduction to Linear Regression Analysis.* Wiley series in probability and mathematical statistics. John Wiley & Sons, 1982. ISBN 0-471-05850-5.

[57] J. Moody and C. Darken. Learning with localized receptive fields. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proceedings of the 1988 Connectionist Models Summer School*, pages 133–143. Carnegie Mellon University, Morgan Kaufmann, June 1988.

[58] NeuralWare, Inc., Pittsburgh. *Using NeuralWorks*, 1993. A Tutorial for NeuralWorks Professional II/PLUS and NeuralWorks Explorer.

[59] B. Nicholls, J. W. Shepherd, and J. Shepherd. Interactive searching of facial image databases. In *Proceedings of the SPIE conference on Investigative and Trial Image Processing*, volume 2567, pages 228–237, San Diego, July 1995.

[60] N. R. Pal and S. K. Pal. A review on image segmentation techniques. *Pattern Recognition*, 26:1277–1294, 1993.

[61] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, February 1990.

[62] N. D. Porter. Neural networks for the classification of facial features. In A. B. Bulsari, S. Kallio, and D. Tsaptsinos, editors, *Solving Engineering Problems with Neural Networks*, pages 293–296, London, June 1996.

[63] R. H. Pugmire, Hodgson R. M., and Chaplin R. I. The properties and training of a neural network based universal window filter(UWF). In *Image Processing and its Applications*, pages 642–646, Edinburgh, July 1995. Herriot-Watt University.

[64] H. T. F. Rhodes. *Alphonse Bertillion - Father of Scientific Detection.* Harrap & Co,, 1956.

[65] S. Rimmer. *Bit-mapped graphics.* McGraw-Hill, 2nd edition, 1992.

[66] F. Rosenblatt. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan Press, Washington, 1961.

[67] M. Rosenblum, Y. Yacoob, and L. Davis. Human emotion recognition from motion using a radial basis function network architecture. In *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pages 43–49, Austin, TX, November 1994.

[68] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 1. MIT Press, Cambridge, 1986.

[69] E. Saber, A. M. Tekalp, and K. Knox. Automatic image annotation using adaptive colour classification. *Graphical Models and Image Processing*, 58(2):115–126, March 1996.

[70] A. Samal and P. A. Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25(1):65–77, 1992.

[71] P. Seitz and M. Bischel. "the digital doorkeeper" - Automatic face recognition with the computer. In *Proceedings of the 25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, pages 77–83, Taipei, Taiwan, October 1991.

[72] J. W. Shepherd. Specification of the FRAME coding procedure with physical measurement of photographic images. University of Aberdeen, Department of Psychology.

[73] J. W. Shepherd. An interactive computer system for retreiving faces. In *Aspects of Face Processing*, pages 398–409. NATO, July 1985.

[74] J. W. Shepherd. Analysis of parameter values and search outcomes for the "FACES" system implemented in Lancashire constabulary. Technical report, Department of Psychology, University of Aberdeen, March 1993. Home Office Contract Report 912021901.

[75] J. W. Shepherd and J. B. Deregowski. Races and faces - a comparison of the responses of Africans and Europeans to faces of the same and different races. *British Journal of Social Psychology*, 20:125–133, 1981.

[76] J. W. Shepherd, H. D. Ellis, and Davies G. M. Perceiving and remebering faces. Technical report, Department of Psychology, University of Aberdeen, 1977. Home Office Contract Report POL/73/1675/24/1.

[77] Hsun-Yu Sidney, Yong Qiao, and Demetri Psaltis. Optical network for real-time face recognition. *Applied Optics*, 32(26):5026–5035, September 1993.

[78] L. Sirovich and M. Kirby. Low-dimennsional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, March 1987.

[79] R. B. Starkey. Facial recognition for police purposes using digital neural networks. Master's thesis, Imperial College of Science, Technology and Medicine, 1991.

[80] T. J. Stonham. Practical face recognition and verification with WISARD. In *Aspects of Face Processing*, pages 426–441. NATO, July 1985.

[81] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[82] K. P. Unnikrishnan and K. P. Venugopal. Alopex: A correlation-based learning algorithm for feedforward and recurrent neural networks. *Neural Computation*, 6(3):469–490, 1994.

[83] S. Usui, S. Nakauchi, and S. Miyake. Neural network model of colour vision. *Proceedings of the 11th Annual International Conference of IEEE Engineering in Medicine and Biology Society*, pages 2044–2045, June 1989.

[84] E. Verity. *Colour Observed*, chapter 2, Classifying Colour, pages 4–18. Macmillan Press, London, 1980.

[85] J. M. Vincent, D. J. Myers, and R. A. Hutchinson. Image feature location in multi-resolution imagesg using a hierarchy of multilayer perceptrons. In R. Linggard, D. J. Myers, and C. Nightingale, editors, *Neural Networks for Vision, Speach and Natural Language*, volume 1 of *BT Telecommunications Series*, chapter 1, pages 13–29. Chapman Hall, 1992.

[86] J. M. Vincent, J. B. Waite, and D. J. Myers. Automatic location of visual features by a system of multilayered perceptrons. *IEE Proceedings part F*, 139(6):405–412, December 1992.

[87] P. J. Werbos. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences.* PhD thesis, Harvard University, 1974.

[88] C. J. Wu and J. S. Huang. Human face profile recognition by computer. *Pattern Recognition*, 23:255–260, 1990.

[89] Y. Yacoob, H. Lam, and L. S. Davis. Recognizing faces showing expressions. In *International Workshop on Automatic Face- and Gesture-Recognition*, Zurich, 1995.

[90] A. L. Yuille, P. W. Hallinan, and D. S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

[91] J. Zhao, G. Kearney, and A. Soper. Classifying expressions by cascade-correlation neural network. *Neural Computing and Applications*, 3:113–124, 1995.

Bibliography

Bianchini, M., Frasconi, P., and Gori, M. Learning without local minima in radial basis funcion networks. *IEEE Transactions on Neural Networks*, 6(3):749–756, 1995.

Chapman, W. A. *Mastering C Programming.* Macmillan Master Series. Macmillan Press, 1991. ISBN 0-333-49842-9.

Draper, N. R. and Smith, H. *Applied Regression Analysis.* Wiley series in probability and mathematical statistics. John Wiley & Sons, 1981. ISBN 0-471-02995-5.

Ellis, H. D., Jeeves, M. A., and Young, A. W., editors. *Proceedings of the NATO Advanced Research Workshop on "Aspects of Face Processing"*, Aberdeen, June 29 – July 4 1985.

Fahlman, S. E. An empirical study of learning speed in back-propagation networks. Technical Report CMU-CS-88-162, CMU, September 1988.

Fukunaga, K. *Introduction to Statistical Pattern Recognition.* Computer Science and Scientific Computing. Academic Press Ltd., 2nd edition, 1990. ISBN 0-12-269851-7.

Gnanadesikan, R. *Methods for Statistical Data Analysis of Multivariate Observations.* Wiley series in probability and mathematical statistics. John Wiley & Son, 1977. ISBN 0-471-30845-5.

Goossens, M., Mittelbach, F., and Samarin, A. *The LaTeX Companion.* Addison-Wesley, 1994. ISBN 0-201-54199-8.

Graybill, F. A. and Iyer, H. K. *Regression Analysis: Concepts and Applications.* Wadsworth Publishing Company, 1994. ISBN 0-534-19869-4.

Heller, D. *XView Programming Manual*, volume 7 of *The X Window System*. O'Reilly & Associates, Inc., 1990. ISBN 0-937175-52-8.

Jordan, M. I. Why the logistic function? A tutorial discussion on probablilities and neural networks. Computational Cognitive Science Technical Report 9503, Massachusetts Institute of Technology, August 1995.

Kamel, M. S., Shen, H. C., Wong, A. K. C., and Campeanu, R. I. System for the recognition of human faces. *IBM Systems Journal*, 32(2), 1993.

Leow, W. K. and Miikkulainen, R. Representing visual schemas in neural networks for scene analysis. In *Proceedings of the International Conference on Neural Networks*, volume III, pages 1612–1217, 1993.

O'Muircheartaigh, C. A. and Payne, C. *Exploring Data Structures*, volume 1 of *The Analysis of Survey Data*. John Wiley & Sons, 1977. ISBN 0-471-01706-X.

Robertson, G. amd Craw, I. The face value of test data. *Image Processing*, 5(4):14–17, 1994.